YOUR

COMMODORE

AUGUST 1988 £1.20

GEOS – Being used at last!



Wordprocessors
Which one's for you?

Expert V3.2
As fast as a parallel DOS

Writing Structured Programs

UNBEATABLE PROGRAMS:

- Plus/4 SIMPLE
- C64/C128 Terminal
- Tape Organiser
- Educating SID
- A Short Interlude

Games Reviewed: Karnov ▲ Patton vs Rommel
Wizard Warz ▲ Infiltrator II ▲ Jinxs

TOTAL BACKUP POWER
Now ACTION REPLAY

NOW EVEN MORE POWERFUL, MORE FRIENDLY AND WILL

ONLY £29·99 POST FREE

Action Replay works by taking a *'SNAPSHOT'* of the program in memory so it doesn't matter how the program was loaded – from tape or disk – at normal or turbo speed.

- **WARP 25. THE WORLD'S FASTEST SERIAL DISK TURBO – NOW EVEN FASTER! LOADS 200 BLOCKS IN 6 SECONDS! 240 BLOCKS IN 7 SECONDS!** – that's even faster than some parallel systems. Built into the cartridge – no extra hardware or software required. Includes supercast, warpsave, scratch, filecopy/convert. Integrated with normal disk turbo for supercompatibility – auto detects warp or normal format so no special load commands are required. Turbo and Warp 25 speed are entirely independent of the cartridge with SUPERBOOT.
- **SIMPLE TO USE:** Just press the button and make a complete backup – tape to disk, tape to tape, disc to tape, disk to disk. THE PROCESS IS AUTOMATIC – JUST GIVE THE BACKUP A NAME.
- **TURBO RELOAD.** All backups will reload at turbo speed, COMPLETELY INDEPENDENTLY OF THE CARTRIDGE.
- **SPRITE KILLER.** Make yourself invincible. Disable sprite collisions – works with many programs.
- **PRINTER DUMP.** Freeze any game and print out the screen. Eg. loading picture, high score screen etc. Works with most printers. MPS 801, 803, Star, Epson etc. Double size, 16 shades, reverse print option. Very versatile – no user knowledge required.
- **PICTURE SAVE.** Save any Hires multicolour screen to disk at the push of a button. Compatible with Blazing Paddles, Koala, Artist 64, Image System etc.
- **SPRITE MONITOR.** Unique Sprite monitors allows you to freeze the action and view all the sprites, watch the sprite animation, save or delete any sprite. Load sprites from one game into another to make customised programs.
- **POKES/CHEAT MODE.** Press the button and enter those pokes for extra lives etc., then restart the program or make a backup. Ideal for custom games.
- **MULTISTAGE TRANSFER.** Even transfers multistage programs from tape to disk. The extra parts fast load – a unique feature. Enhancement disk available for non standard multi-loaders (see below).
- **SUPER COMPACTOR.** Ultra efficient program compaction techniques. Each program saved as a single file. 3 programs per disk side – 6 programs per disk, if you use both sides.
- **TEXT MODIFY.** Change title screens, high score screens etc. Put your own name into a game then restart it or make a backup to tape or disk. Very simple to use.
- **MONITOR.** Full feature 'Floating' type MACHINE CODE MONITOR. All standard features plus many more:- assemble, disassemble, hex dump, interpret, transfer, compare, fill, hunt, number conversion, bank switching, relocate, load/save etc. Uses no memory. Full printer support.
- **DISK MONITOR.** A special monitor for use on the RAM inside your disk drive. All the usual commands – a useful hacking tool.
- **WHOLE DISK COPY.** Copy a full unprotected disk in under two minutes with only one drive.
- **FAST FILE COPY.** Works with standard and Warp 25 files of up to 249 blocks. Converts formats to and from Warp 25.
- **FAST FORMAT.** Under 20 seconds.
- **TOOLKIT COMMANDS.** A whole range of useful new commands including: AUTO LINE NUMBERING, DELETE, MERGE, APPEND, OLD, LINESAVE, etc., PRINTERLISTER – list any program (including directory) directly from disk to printer or screen without corrupting memory.
- **REDEFINED FUNCTION KEYS.** Single stroke commands for operation of many common commands including: LOAD, SAVE, DIR. Load from directory – no need to type in filename.
- **TAPE TURBO.** Designed to make turbo load/save for your own programs. No screen blanking during loading.



WHAT THE REVIEWERS SAID
"I'm stunned, amazed and totally impressed. This is easily the best value for money cartridge. The cartridge king!
Commodore Disk User

REMEMBER all features are built in and available at the touch of a key. All features work with both **TAPE** and **DISK.** (Except multipart transfer & disk file utility).

ACTION REPLAY ENHANCEMENT DISK

The biggest and best collection of special parameters and file copy programs for transferring non-standard multi-load tapes to disk – games like **LAST NINJA, CALIFORNIA GAMES, LEADERBOARD, DRAGON'S LAIR** – SEVENTY titles in all. Almost all major titles covered. Latest edition includes **COMBAT SCHOOL, PLATOON, PREDATOR, GAUNTLET II, TESTDRIVE, SKATE OR DIE, APOLLO 18, THE TRAIN** and many more. Cheats for infinite time, lives etc. The **GRAPHIC SLIDESHOW** – latest edition displays multicolour pictures or loading screens saved by Action Replay or any major Art Package – Blazing Paddles, Koala, Advanced Art Studio, Artist 64 etc. Lots of fun. Only **£7.99**. Upgrades – send £3.00 plus old disk.

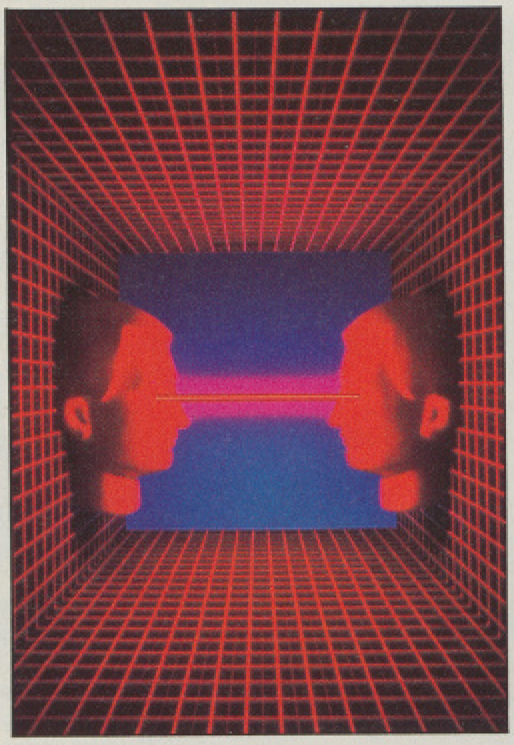


Editor: Stuart Cooke
Deputy Editor: Sue Joyce
Technical Editor: Fred Reid
Advertisement Manager: Paul Kavanagh
Advertisement Copy Control: Andrew Selwood
Origination: Ebony Typesetting
Artist: Alan Batchelor
Designer: Neil Sweetman

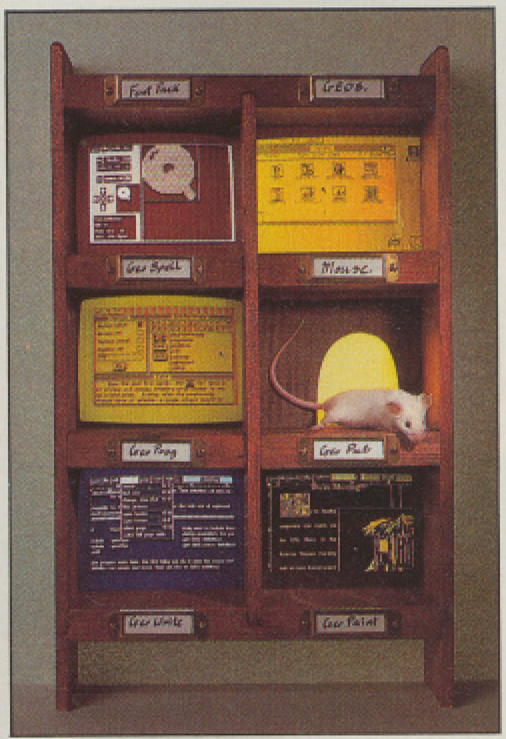
Your Commodore incorporating Your 64 is a monthly magazine appearing on the first Friday of each month. Your Amiga is published every second month within the pages of Your Commodore. Argus Specialist Publications Limited Editorial & Advertisement Office, Your Commodore, No 1 Golden Square, London W1R 3AB. Telephone: 01-437 0626 Telex: 8811896.
Subscription rates upon application to Your Commodore Subscriptions Department, Infonet Ltd, 5 River Park Estate, Berkhamsted, Herts, HP4 1HL. U.S.A. Subscription Agent: Wise Owl Worldwide Publications, 4314 West 238th Street, Torrance CA 90505 U.S.A.

ABC
MEMBER OF THE AUDIT BUREAU OF CIRCULATIONS

FEATURES



Communications



Geos goes to work

UTILITIES

REGULARS

VOLUME 4
NUMBER 11

ARGUS
PRESS
GROUP

SEPT ISSUE AVAILABLE 5th AUG 1988

 Distribution SM Distribution, 6 Leigham Court Road, London SW16 2PG. Printed by Chase Web, Plymouth.

ISSN
0269-8277

DATA STATEMENTS

Commodore Beats Amstrad!

COMMODORE HAVE announced a dramatic price cut on their PC1 personal computer, from £499.99 to £369.99, including mono monitor, Able software and VAT. This puts the Commodore PC1 some £30 more attractive than Amstrad's comparable machine, and makes it the cheapest PC 'clone' on the market in this country! The PC1 uses an 8088 processor running at 4.77MHz, and comes with the usual 512K RAM, expandable to 640K. Ready to run business software, MS-DOS 3.2 and a built in 5¼" 360K floppy complete the picture.

The PC1 is probably one of the smallest desktop micros around, measuring just 13 inches square, it should prove popular with those short of desk space, the Able software includes wordprocessing, spreadsheet and database management software.

The ball's in your court, Amstrad.....

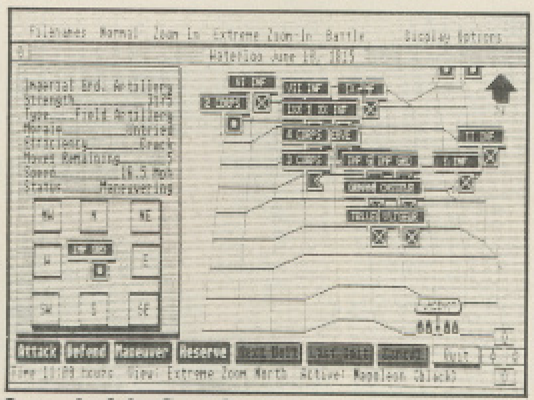
Busby gets Busy on Amiga

TELECOM SOFT have been busy this month with five major game releases for the Amiga from Rainbird and another two from Firebird. Rainbird's jewel in the crown is undoubtedly Starglider 2, the sequel to Jez San's blockbuster, Starglider. Your mission (should you choose to accept it) is to annihilate the Egron army once and for all, and destroy the beam projector on their home planet, Novenia. The odds are stacked against you, you can only enter Egron system unarmed. On top of that, your Hallucitron projector is on the blink and it's generating some pretty hairy images of hideous monsters and mutated creatures.

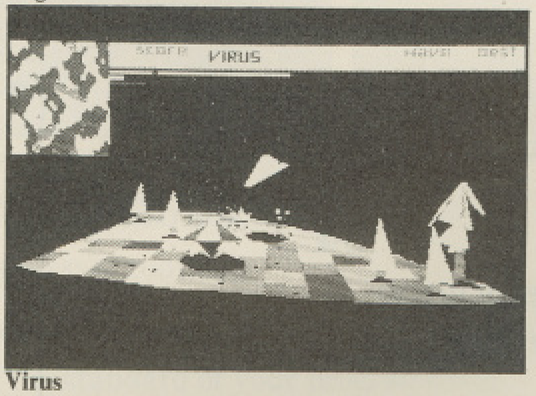
Next up, Carrier Command, a strategic shoot 'em up on the high seas. Take the helm of a futuristic aircraft carrier complete with fighter planes and amphibious assault craft. Whether you attack the enemy installations with planes, tanks or both, you'll have to run the gauntlet of suface to surface and surface to air missiles, lasers and enemy tanks and planes.

Legend of the Sword, a mega adventure game, tells of the chronicles of Anar, a mystical sword and a shield whose magical aura protected the inhabitants of Anar for many centuries. But now Anar has been plunged into a state of turmoil and fear by an invading force of telephone engineers – oops, mutated humanoids under the evil wizard Suzar. Only with the combined powers of the sword and shield is it possible to defeat the dark forces.

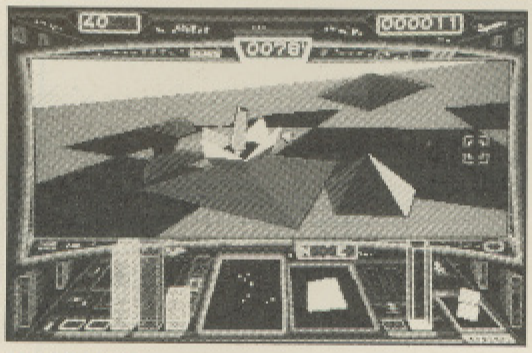
Where did Napoleon keep his armies? Up his sleevies! This could be your chance to find out for yourself! The Universal Military Simulator is an entirely new concept in war games. Re-enact some of the world's greatest military conflicts including the battle of Hastings, Marston Moor, Waterloo and Gettysburg against the computer or with a friend. The graphics are rumoured to be superlative in 3-D, view from any angle, or zoom in for a spot-check on your troops. You're not stuck with 'real' battles, you can create your own situations, maps and armies, even match heros from different time zones! How would you like to pit Monty's Desert Rats against Alexander the Great?



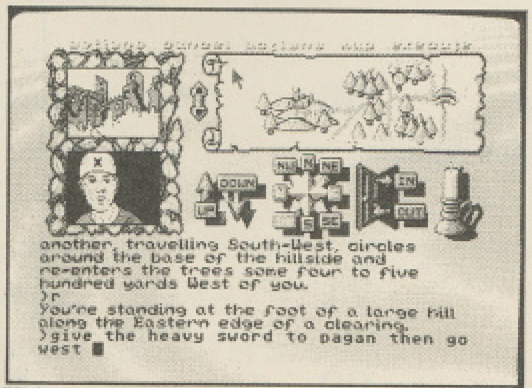
Legend of the Sword



Virus



Star Glider 2



Universal Military Simulator

Firebird's first contribution this month is a superfast 3-D shoot 'em up by David Braben, co-author of Elite. Virus is actually a conversion of Zarch, currently knocking 'em dead on the Archimedes. Invading aliens are attacking your planet, polluting the surface with a deadly virus. Your job is to destroy the alien craft and thus prevent the spread of the virus. Having seen the Archimedes version, I can't wait to see it on the Amiga.

Also from Firebird is Whirligig, a space battle extravaganza. Whirligig tells the story of ships with brains, kept gleaming and mechanically healthy by slaves – small humanoid bio-things, skurrying around the ships to tend to their every need. You are determined to steal the to be born (?) and boldly seek out brave new worlds. The whirligig is actually an interspacial shortcut to other worlds and times, but you'll need to blast your way through fleets of alien spacecraft, picking up new weapons as you go........

***Touchline:** Starglider 2, Carrier Command, Legend of the Sword and Universal Military Simulator, all at £24.95 each, from Rainbird. Virus and Whirligig cost £19.95 each, from Firebird.*

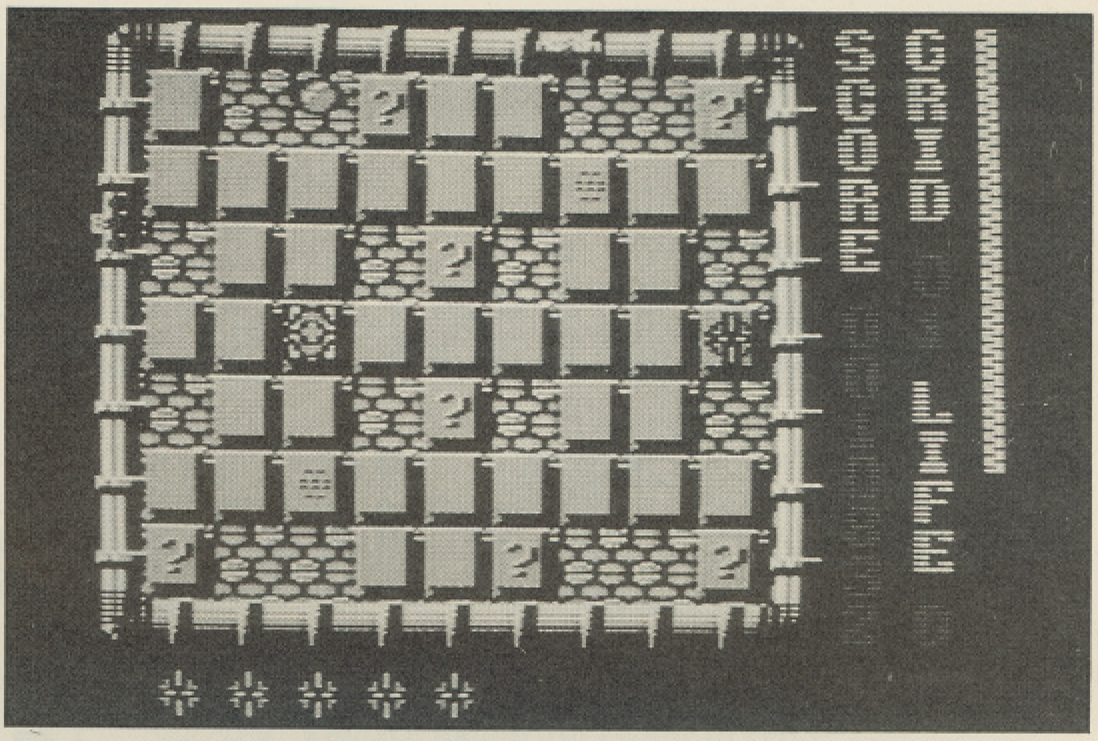
DATA STATEMENTS

Commodore v Chelsea

THOUGH COMMODORE are to be commended for their sponsorship of Chelsea FC, recent events must pose a dilemma for both company and club. The appalling behaviour of Chelsea supporters at the recent playoffs against Middlesborough not only reflects badly on both of the football clubs but also on the companies who support them.

The television coverage showed the full horror of the incident as Chelsea supporters surged across the pitch towards the Middlesborough supporters. Projectiles, punches and abuse were thrown with equal abandon as police battled to control the melee. The greater tragedy is that this incident not only discredits football in this country but will also feature strongly as the European debate on the re-entry of English clubs into full international competition rages on. During this debate the video recording of the incident will no doubt be shown and reshown with Commodore's name featuring strongly and receiving some of the tarnish. Subliminal though this inference may be, it will exist. Can Commodore afford to be connected with a team who is supported by a strong hooligan element?

Chelsea and other League clubs maintain that they are doing everything humanly possible to eradicate the violent element from the sport but this is blatantly not working. Whether Commodore can afford to maintain their support is their decision. Chelsea cannot be held totally responsible for their fans at all times but the sponsors must ensure that more is done both for the reputation of English football and the honour of the company name.



Oops! News from The Big Apple

'Oops!', a thoroughly addictive arcade style game, the first offering from London's newest software house is about to hit the streets. The Big Apple label plan to release several more titles in the near future, watch out for Delphian and Neutron on all formats in the near future. Oops! is available mid-June on the C64 (£9.95) and Amiga (£19.95).

Touchline:

The Big Apple Entertainment Co. ***Tel:*** *01-368 5545.*

Programmer's Olympics

THE FIRST COMPUTER Olympiad is all set for August, at London's prestigious Park Lane Hotel. For the first time, you will be able to see machine pitted against machine, program against program and journalists against the bar! This unique event will feature tournaments for chess, bridge, backgammon, draughts, poker, and many other 'thinking' games, the human operators doing no more than telling their own computers what moves have been made by their opponents.

The first London Conference on Computer Games will take place as part of the Olympiad chaired by Professor Tony Marsland from the computing science department of the University of Alberta, Canada, papers will be invited on all aspects of programming computers to play 'thinking' games.

The Computer Olympiad is organised by International Chess Master David Levy, who is president of the International Computer Chess Association. Anyone wanting further information on the event should send a large stamped addressed envelope to: Computer Olympiad, 11 Loudoun Road, London NW8 0LP.

Electronic Arts in $7.3 Million Lawsuit

BETHESDA SOFTWORKS, developer of Gridiorn! (an American football simulator) is accusing California based software publisher and distributor Electronic Arts of forcing their game off the market. Bethesda claim that Electronic Arts offered them firm development and distribution contracts in order to promote their own American football game, 'John Madden Football'.

Under the 1987 contract, Electronic Arts gained exclusive marketing rights to the Amiga and Atari ST versions of Gridiorn! and prohibited Bethesda from developing it for any other computer, according to documents filed in the California federal court.

As a condition of the marketing agreement, Electronic Arts also called for major parts of Gridiorn! to be incorporated into a new game featuring former Oakland Raiders coach and current CBS commentator, John Madden, according to Bethesda's suit.

In a nutshell, Bethesda Software are accusing Electronic Arts of conning them into designing Gridiorn, and then using the game as a basis for their own 'John Madden Football'. Would anyone from Electronic Arts care to reply?

Win Expert Prizes from Trilogic

Spot the differences and you could be the proud owner of the Expert cartridge, the Voice Digitiser and the Data Sector Doctor

We've teamed up with Trilogic for this month's competition. The first entry picked out of the hat will win the Expert cartridge, the Voice Digitiser and the Data Sector Doctor. The nine runners up will each reveive a copy of either the Voice Digitiser or the Data Sector Doctor. (Please state preference on entry coupon.)

Trilogic Entry Coupon

Name ....

Address ....

.... Postcode .....

Number of differences found ......

Voice Digitiser ☐

Data Sector Dector ☐
(please tick)
Closing date: 31st August 1988

Post to: Trilogic Competition
Your Commdore
1 Golden Square
London W1R 3AB





DIMENSION computers l.t.d.

27/29 High Street
Leicester LE1 4FP
Tel: (0533) 517479/21874

THE MIDLAND'S LARGEST COMPUTER STORE

CANON PJ1080A COLOUR INK JET
* Virtually silent operation
* Seven colour text and high resolution colour graphics printing (640 dots per line)
* Uses plain, single sheet or roll paper
* Supplied with AMIGA printer driver software £499.95

AMIGA BUSINESS PACK
* Amiga A500 computer
* 1084 high resolution colour monitor
* Citizen 120D printer
* Monitor and printer cables
* The works-integrated spreadsheet, wordprocessor and database
* Transformer-PC Emulator £779.00

302C 1Mb A500 3.5" External Drive £99.95
Cumana 1Mb 3.5" External Drive £109.95
Internal A2000 2nd Drive £79.95
Amiga Midi Interface £49.95
Digiview Colour Digitiser £139.95
Supra 20Mb A500 Hard Disc including Diskmaster Software £649.00
2Mb Internal RAM Expansion £449.00
A500 512K RAM Upgrade £109.95
Future Sound Digitiser, including microphone £169.95
Perfect Sound Digitiser, with stereo recording facility £74.95
Audiomaster Sampling Software £39.95
Sonix Synthesis Software £49.95
Prowrite V2 Wordprocessor £79.95

AMIGA RANGE

AMIGA A2000 PACK OFFER
Amiga A2000 1Mb with Mouse, Workbench, Workbench Extras, GW Basic and 20Mb A2090/2092 Hard Disc
With 1081/1084 Monitor .... £1379.00 + VAT
Without Monitor .... £1169.00 + VAT

AMIGA A500 PACK 1
with Mouse, Workbench, Basic, DPAINT, TV Modulator, 40 Disc Storage Box, Dust Cover and 10 Games/Utility Discs £415.00

AMIGA A500 PACK 2
with Mouse, Workbench, Basic and DPAINT Software £379.00

COMMODORE 1084 COLOUR MONITOR
High RES, Non Reflective Screen £229.00

AMIGA A500 with PHILIPS 8833 HIGH RES COLOUR MONITOR with STEREO SOUND OUTPUT, Mouse, Workbench, Basic, plus DPAINT Software £629.00

AMIGA A500 with COMMODORE 1084 HIGH RES COLOUR MONITOR Mouse, Workbench, Basic, plus DPAINT Software £599.00

STAR LC10 COLOUR DOT MATRIX
* 120 cps Draft, 30 cps NLQ
* 6 Resident Fonts and 6 Print Pitches
* Draft and NLQ in all Fonts and Colours
* 7 Colours plus Multiple Shade Graphics

Includes AMIGA Printer Lead £229.95

THE WORKS
Spreadsheet, Wordprocessor and Database £69.95

PHILIPS 8833 COLOUR MONITOR £259.00

LIMITED AMIGA SOFTWARE OFFERS

GRAPHICS PACK
Express Paint £59.95
Pageflipper Animator £34.95
IntroCad CAD System £59.95
Normallly £154.85 **OFFER PRICE** £99.95

DESKTOP PUBLISHING PACKAGE
Vizawrite Wordprocessor £99.95
Citydesk Desktop Publisher £99.95
Normally £199.90 **OFFER PRICE** £149.95

BUSINESS PACK
Vizawrite Wordprocessor £99.95
Logistix Spreadsheet £99.95
Superbase Personal Database £99.95
Normally £299.85 **OFFER PRICE** £199.95

IMAGE PROCESSING PACK
Digipic Digitiser £343.85
Pixmate Image Processor £49.95
Baxall Video Camera £229.95
Normally £623.75 **OFFER PRICE** £499.95

THE NEW STAR LC10 DOT MATRIX PRINTER
* 120 cps Draft, 30 cps NLQ
* Friction and Tractor Feed
* Paper Parking Facility
* 6 Resident Fonts and 6 Different Print Pitches

Selectable from Front Mode Select Panel
Commodore Serial or Centronics £199.95
Commodore 64/128 Colour Version £229.95
Including FREE GEOS worth £26.95 with CBM version

1541C DISC DRIVE £169.95
OCEANIC OC 118N £129.95
Commodore Compatible Disc Drive
Both include GEOS, plus either:
Software Pack A
6 Disc Adventure Games plus 5 Disc Arcade Games
Software Pack B
Easyfile Database, and Future Finance Planning Package

CBM 128 Ram Expansion 128K £99.95
CBM 128 Ram Expansion 512K £149.95

SUPER G
CBM to Centronics Printer Interface £34.95
FINAL CARTRIDGE 3 .... £39.95

COMMODORE 560P
2-colour Dot Matrix Printer, takes 2¼" roll paper £49.95 £29.95

MPS 803 TRACTOR FEED UNIT £29.95

CBM 64/128 MUSIC EXPANSION SYSTEM
Including Keyboard, Synthesiser and Software £75.00

COMPOSER/EDITOR SOFTWARE
For Music Expansion System, Tape or Disc £24.95

ACTION REPLAY Mk IV
Tape to Disc Back-up Cartridge £29.95

NEOS MOUSE
With CHEESE Software, Tape or Disc £24.95

COMMODORE C2N DATA RECORDER £29.95

PRINTER RIBBON OFFER
MPS 801, 802, Citizen 120D and Star NL10 Ribbons £5.95
MPS 803 & MPS 1000 .... £4.95
DPS 1101 Multi Strike £3.50
BUY FOUR AND GET ONE FREE!

REMARKABLE COMMODORE PC1 OFFER
Unbeatably priced PC/XT compatible, including extremely compact base unit, detachable keyboard and monitor. Compatible with all major PC/XT software, the PC1 now represents the lowest price for specification PC compatible in the UK.
* 8088/4.77MHz Processor
* 512K RAM
* Serial and Parallel Ports
* MSDOS 3.2 and GW Basic included

For a limited period, the PC1 also includes ABLE ONE Integrated Spreadsheet, Wordprocessor and Database package (usual price £113.00)
PC1 with Mono Monitor .... £299.99 + VAT
PC1 with CGA Colour Monitor .... £449.00 + VAT
PC1 Package without Monitor.
Compatible with CBM 1084 and most CGA/Hercules monitors .... £269.00 + VAT

CITIZEN 120D DOT MATRIX PRINTER
Commodore version (with 64/128 Printer Lead) £159.95
Centronics version (with AMIGA Printer Lead) £159.95
Commodore 1901 Colour Monitor £219.95
Philips 64/128 Green Screen Monitor £89.95
MPS 1200P Dot Matrix Printer, Centronics £219.95
MPS 1500C 7 Colour Dot Matrix, Centronics £339.95
1581 3½" Disc Drive including 10 x 3½" Discs £199.95

PRECISION
Superscript 64 .... £24.95
Superbase 64/+4 .... £24.95
Superscript 64 & Superbase 64 .... £44.95
Superscript 128 .... £29.95
Superbase 128 .... £29.95
Superscript 128 & Superbase 128 .... £54.95
Superbase, The Book £11.95

GEOS Version 1.3
CBM 64/128 Operating System
Including: Desktop, Geowrite, Geopaint, and Desk Accessories, Disc
CBM64 .... £26.95
CBM128 .... £34.95

GEODEX/DESKPACK
Combined Database with powerful Mailmerge function. Also contains Artgrabber Graphic Images, Calendar and Icon Editor £21.95

GEOCALC Spreadsheet £24.95
GEOCALC 128 .... £34.95
GEOPUBLISH Desktop Publisher £34.95
WRITERS WORKSHOP £21.95
WRITERS WORKSHOP 128 .... £34.95
FONT PACK 1 .... £16.95
GEOFILE Database £24.95
GEOFILE 128 .... £3495
GEOSPELL Spellcheker £16.95

I CONTROLLER*
Geos-compatible Joystick. Fits onto keyboard £12.95
*FREE OF CHARGE WITH PURCHASES OF TWO OR MORE GEOS PACKAGES

CBM 64/128 and PLUS 4 COMPANIES PACK
Sales/Purchase/Nominal Ledgers with Invoicing and Stock Control, Disc £24.95

PERSONAL CALLERS AND EXPORT ENQUIRIES WELCOME
ALL OFFERS STRICTLY SUBJECT TO AVAILABILITY
ALL PRICES SUBJECT TO CHANGE WITHOUT NOTICE
ALL PRICES INCLUDE VAT UNLESS OTHERWISE STATED

To order, either send cheque/P.O. with coupon or ring **(0533) 517479/21874** with Access/Barclaycard number. Please allow 7 working days for delivery from receipt of order or cheque clearance.
Please add £1.00 towards carriage and packing on any software order.
Hardware orders add £1.00 under £65.00 value, and £5.00 over £65.00 value.
Printer orders add £5.00 towards carriage.

Send to: Dept. No. YC 8, Dimension Computers Ltd., 27/29 High Street, Leicester LE1 4FP.
Tel: (0533) 517479/21874 Fax: (0533) 511638

VISA

Item	Quantity	Price
....		
....		
	Total	

Name..........
Address..........
....
Telephone:..........

The Graphic Environment Operating System, that was once set to be bundled with the Commodore 64C has at last come of age

GEOS

In its original format GEOS consisted of a single disk that contained the Amiga style desktop as well as GeoWrite and GeoPaint applications programs. However, these were incredibly basic programs and were little more than demonstrations of what could be done with a GEOS environment.

In December 1986 Berkley Softworks, the author of GEOS, released GEOS 1.2 with updated versions of GeoWrite and GeoPaint and also the Writer's Workshop (turned GeoWrite into a full wordprocessor), Fontpack 1 (20 new fonts), GeoDEX (card index system), Deskpack (calender, graphics grabber and icon editor) applications which were followed later by a full database program, GeoFile and GeoCalc the spreadsheet.

Armed with all these a dedicated user could turn GEOS into a workable system, however you were soon confronted by its limitations.

Now, four major factors have come together which will spread the use and variety of GEOS. Firstly, Berkley Softworks has made further improvements to existing packages and has added more to the range including GeoPublish, GeoSpell and GeoProgrammer. Secondly, this expanding GEOS range is now being distributed in greater quantities through a joint venture with Microprose UK which has meant dramatic cuts in costs. Thirdly, GEOS is being bundled with new disk drives such as those supplied by Evesham Micros and finally in this article we review the first third party books and software.

What is GEOS?

GEOS is a C64 and now C128 disk operating system that attempts to mimic the icons, pull down menus and workbench system pioneered on machines such as the Apple Macintosh, Atari ST and Amiga. It's a "You get what you see" or "if you want it, click it" system where a click of a mouse or joystick button can load in files, select options or swap disks.

Each package or program that you add to the system must be installed or keyed into your own copy of GEOS. This not only prevents piracy but is the beginning of a mass file copy session in which you create work disks containing all the programs, files and type fonts you will need.

This is important as GEOS is a fully integrated system in which you could create graphics in GeoPaint or grab them from Print Shop, Print Master or Newsroom disks and mix them with text in GeoWrite documents which can then be used for a standard letter sent out by GeoMerge to some or all of the names and addresses stored in a GeoFile database or GeoDex card file.

Upgrades

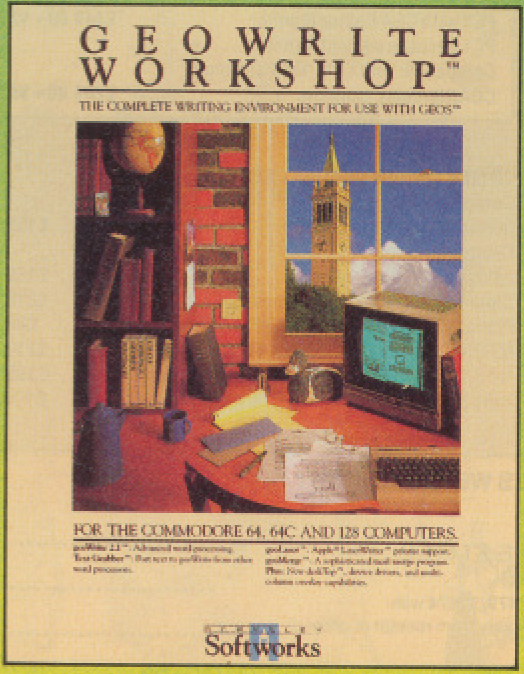
1988 sees a more streamlined GEOS system with many of the old upgrades now included in the new standard packs. GEOS itself is now supplied on two disks so you now have an automatic backup copy (once it's been installed) and a built-in disk copy utility to ease the creation of work disks. This is all backed up by an improved and enlarged manual.

The new GEOS system and all subsequent application programs including the ones mentioned below include the 1.3 version of desktop as standard with its faster disk access and keyboard shortcuts. For frequently used commands such as opening or closing disks, pressing two keys is a lot faster than selecting options from pull down menus. In true GEOS style the keyboard shortcuts aren't buried away in a manual but are alongside the appropriate function or command in the pull down menu.

Desktop 1.3 also includes more printer drivers so that more printers and interfaces can now be used to printout GEOS text and graphics. A full table and guide to help you install the right printer driver are included in the manual.

Deskpack Plus is the updated version of Deskpack and includes the original combination of Calender, Icon Editor, BlackJack and Graphics Grabber but now also includes the GeoDex and GeoMerge programs that were sold separately as GeoDex.

Similarly, the old Writer's Workshop has been updated to form the GeoWrite Workshop which is now available in both C64 and C128





Goes to Work

versions and has the additional ability to include customised headers and footers and three types of printout in almost any shape or size.

Expanding the Range

Berkley Softworks has released four new packages to extend the GEOS range – GeoPublish, GeoSpell, Fontpack Plus and GeoProgrammer.

GeoPublish adds a full desktop publisher to the GEOS range and can be used to create newsletters and magazines.

Creating a page or a longer magazine couldn't be easier as each step is described in detail in the manual and demonstrated through worked examples.

Before the task can begin you must design a master page format by deciding the height of the headline, what will appear at the foot of each page and the number and size of the columns of text. This can range from a single column spread across the page or two, three or four thinner columns for a more professional look.

Once you have created master page formats, or selected one of the sample formats included on the GeoPublish disk you're ready to make up the pages.

Switching to Page Layouts mode allows you to box off parts of the page and assign them to either text or graphics. You don't actually have to type in the text, you simply assign a file either created by GeoWrite or converted by the text grabber. GeoPublish will then paint in the text in the space you've allocated with any remaining words carried onto the next page.

Similarly, you can assign a space for graphics and then fill it with anything stored in a GEOS picture album. These can range from GeoPaint masterpieces to clip art "grabbed" from Newsroom, Print Master or Print Shop.

That's only half the story as each box can be moved to anywhere on the screen, altered in size and shape and filled with a picture that can be cut or cropped to fit, reduced to fit or stretched to fill the allocated space.

Whenever you change a page layout, GeoPublish automatically redistributes the text through the pages leaving you to concentrate on the design and layout.

GeoPublish introduces a new command tool to the existing pull down menus and dialogue boxes which usually pose yes/no questions but can be used to select brush widths and fill patterns. This is a toolbox which contains up to 12 icons representing additional commands that are specific to the screen mode you are in and are activated by a simple click of a joystick or mouse button. They are at first confusing and will have you diving for the manual, as you create your first masterpiece but they soon become second nature and you wonder how you ever managed without them.

Now with the pages filled with all the text running neatly around the appropriate graphics you can add a headline (up to 192 point 2½ inches high) or zoom in to edit part of a page.

In edit mode you can add and delete words and even change the font and point size to add cross heads and captions to your text and graphics. You can also use a mini graphics utility to add the final touches such as ruled lines, circles or boxes that can be drawn in a variety of thicknesses and filled in a selection of patterns.

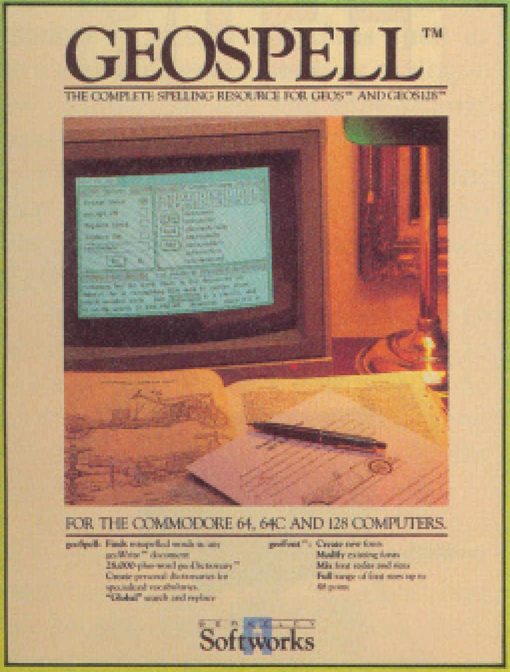
The final result should be a masterpiece, if not you can add or edit it or start again and then print it out on a dot matrix printer or through LaserPublish and a laser printer which will greatly improve its appearance. This isn't as drastic as it sounds as the words and pictures will still be secure in desk files as GeoPublish only decides the order and positions that they are used in.

Fontpack Plus can improve the style and variety of any GEOS document as it includes 43 new fonts and ten of the best from Fontpack 1.

These new fonts include some that consist of symbols instead of letters and numbers so you can easily add scientific and electronic notations as well as trees, musical notes and house contents. In other words everything including the kitchen sink!

If a total of 53 fonts isn't enough, and it should be for most Geowrite documents and to spruce up most GeoPublish pages then you can create your own with GeoFont which is also included on the Fontpack Plus disk.

This works the same as any character editor and can either be used





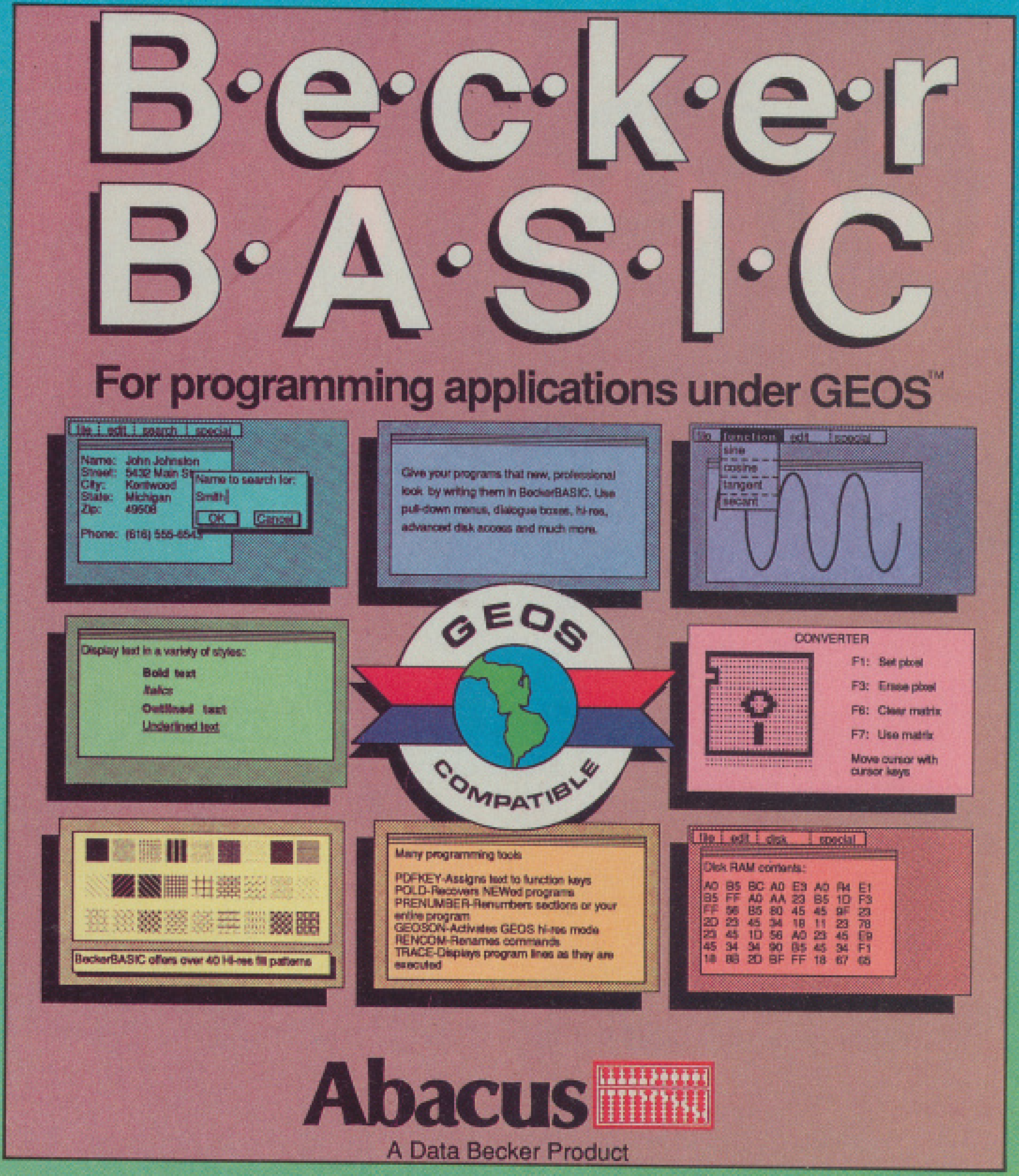
to alter existing fonts or to create your own including pictures which can add character graphics to GEOS.

GeoSpell also includes GeoFont as well as a 28,000 word, upgradable spell checker which can be used to correct even the worst spulling mistakes in any GeoWrite file including any grabbed by the text grabber and converted to GeoWrite format.

Twenty-eight thousand may sound a little low for a dictionary but you can freely add words to user dictionaries and swap between them at will.

GeoSpell makes up for this apparent lack of words through ease of use as whenever it finds a word it doesn't know you have the choice to accept it as it is, use the alphabet keys to search for any word that might be stored in any of your dictionaries or use the FIND key to produce a scrolling list of possible options that can then be used to replace the word as well as every other time that word appears.

The transition from the user friendly end user application GeoSpell to the assembly language development package GeoProgrammer is about the biggest jump you can make.





GeoProgrammer is a scaled down version of Berkley Softwork's own UNIX development system and can be used to create GEOS application programs. The program is supported by a massive 400 page manual that describes in great detail the three parts of the system, GeoAssembler, GeoLinker and GeoDebugger.

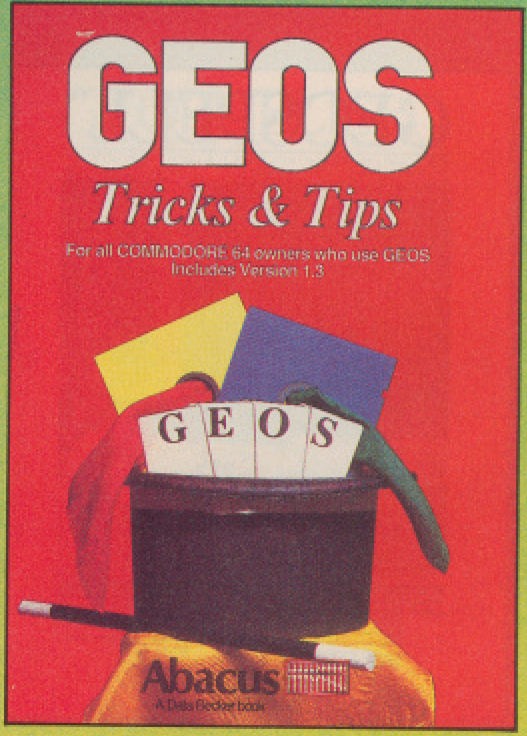
GeoAssembler takes 6502 assembly instructions and creates linkable object files which are then linked by GeoLinker to form either stand alone C64 programs or GEOS applications that can be tested by GeoDebugger which allows you to instantly toggle between the GEOS hi-res screen and the text mode debugging screen.

Each section boasts an impressive range of features such as the support for overlay modules in GeoLinker and GeoDebugger's 80 commands that allow you to assemble, disassemble, single step, display variables and set breakpoints.

GeoAssembler code is written into a normal GeoWrite file so you can add different fonts to make your code easier to understand without affecting the code. For example, you could highlight labels in italics or in 48pt if you really want them to stand out! You can even use the DeskPack Icon Editor to create icons and GeoPaint for graphics which are automatically converted into binary data.

If you don't think you're ready for GeoProgrammer then Becker Basic could be more your style.

This Abacus extended basic breaks new ground as one of the first pieces of third party software and brings the ability to create GEOS programs complete with pull down menus, icons and dialogue boxes in the reach of the Basic programmer.



Becker Basic consists of 273 new commands that are used in the input and testing systems to create a Becker Basic program that can then be run as a GEOS application by double clicking its desktop icon or run on its own through the third part of the program.

Each of these commands can be renamed so their actions can be made clearer. For example, PRINT could become OUTPUT or even PAPER. This may not seem important but a renaming session could save you a lot of time delving in the manual for the right command.

These commands include programming tools such as TRACE, and RENUMBER, structured programming controls IF/THEN/ELSE, REPEAT, WHILE and LOOP, 35 commands to create and move sprites and 18 to create the modulations, alter the filters, change the waveforms and set the envelopes to make beautiful music.

The menus and dialogue boxes that will give your programs that GEOS touch are built and added to your programs through the pull-down menu and dialogue box construction sets.

With GEOS and Becker Basic in memory it is quite remarkable that there is 16K remaining for your program. However, if you add pull down menus and dialogue boxes to your code you'll be left with about 8K as they require a second hi-res bitmap. Conversely, if you avoid hires graphics altogether you would gain an extra 8K but the result wouldn't be a GEOS program.

Creating a pull down menu or dialogue box couldn't be simpler as the construction set generates the code you need and saves it on disk in response to simple questions such as the number of menu items or dialogue box options and the text they should contain.

You can even have sub menus for your pull down menus that can run either horizontally or vertically down or across the screen.

Dialogue boxes can include up to six options ranging from yes or no to which way to go at a junction in an adventure. A simple branch command will then direct the program to the right section of code.

Here at last are two different ways to create GEOS programs, GeoProgrammer for the assembler and Becker Basic for the others. Both offer ease of use and the chance to use GEOS's menus and boxes in their programs. These not only provide two good ways of programming on the C64 but also opens the door for more GEOS applications.

GEOS Books

The Official GEOS Programmers Reference Guide, published by Bantam Computer Books, is the official tome of facts and addresses for programmers who wish to delve into GEOS without the aid of GeoProgrammer.

Armed with this manual the assembler programmer can unlock the GEOS macro routines to create icons, menus, dialogue boxes, fonts as well as new printer interfaces, graphics libraries and multi tasking applications.

Four hundred and fifty pages that can provide the GEOS equivalent to the C64 Programmers reference guide. Having said that, it shouldn't scare off first time assembler programmers as the book describes these routines through labels that are indexed at the end of the book, and therefore can be substituted for the hex equivalent when you start coding.

GEOS Tricks and Tips, from Abacus the company behind Becker Basic, covers the whole spectrum of GEOS from hints and tips for GeoWrite and GeoPaint users, a guide for programmers and listings for three GEOS programs.

The hints and tips cover common sense and shortcuts which make using GEOS programs a lot easier, such as writing your own error messages and creating GeoWrite form letters.

If you tire of the 50 hints and tips crammed into the book you could type in either a converter or font editor which do the same job as GeoFont and the Deskpack's text grabber or tackle Edmon, a machine code monitor to delve into programming and converting GEOS.

This new burst of GEOS activity which has included the development of a GEOS desktop publisher and spellchecker, the release of programming tools backed up with information aimed at every level of user will attract more and more interest in this valuable but underrated operating system.

GEOS has now evolved from a gimmick to mimic 16 bit machines to a system with its own wordprocessor complete with spellchecker, spreadsheet, graphics package, database and DTP package and also has the ability to incorporate others programs, files and pictures into the system.

To paraphrase someone who was actually describing the Amiga (but its equally appropriate here), GEOS was first a child full of promise but couldn't stand on its own, then it was a teenager as it become more organised and useful but still capable of throwing the occasional tantrum; now it has come of age. YC

Supplier: Microprose UK, 2, Market Place, Tetbury, Glouc
Tel: 0666 54326

Becker Basic (with software)	
GEOS Tricks and Tips	£11.95
The Official GEOS Programmers Reference Guide	£17.50
GEOS 1.3	£29.95
GeoCalc	£29.95
GeoPublish	£39.95
GeoWrite Workshop	£24.95
FontPack 1	£19.95
Fontpack Plus	£24.95
GeoProgrammer	£39.95
GeoFile	£29.95
GeoSpell	£19.95
DeskPack Plus	£24.95
GEOS 128	£39.95
GeoWrite Workshop 128	£39.95
GeoCalc 128	£39.95
Geofile 128	£39.95

Keep it Simple

Give your Plus/4 an Amiga style environment

By Mark Everingham

Rather pretentiously this program is called SIMPLE, an acronym for Simple, Icons, Menus, Pointers, Language and Extension. This is an extension to the normal BASIC language allowing you to write programs operating in a similar way to the WIMP and GEM systems but using the keyboard or a joystick in place of a mouse. It consists of a library of thirteen machine code routines called by the usual SYS command which enable you to perform such functions as handling a pointer, drawing windows, and generating pull down menus.

The pointer is a sprite – usually arrow-shaped which forms the heart of the system. It is moved around the screen (known as the Desktop) and used to select functions of the software simply by pointing at words or pictures.

Icons are the pictures mentioned before, representing actions within a program. These actions are performed by moving the pointer onto the picture and pressing the select button.

Windows are areas of the screen separated from the rest of it by a border. All work goes on within them and they may be closed to leave the screen below intact.

Pull-Down Menus are lists of options which are pulled-down from a menu bar at the head of the screen. A reverse video bar is then moved up and down the menu, and the select key pressed to select an option. Then, the menu may be removed from the screen.

Now that we all know what we're talking about, I will describe the different commands of SIMPLE in detail. Also included is a reference table (Figure 1) of start addresses, syntax, and other relevant information.

Commands

Desk does not need any extra parameters and clears the screen to the grey "Checkerboard" pattern commonly used in mouse-driven software. The desk-pattern is stored in the Character Code 64 and this may be redefined to create different desk patterns. For example:

```
10 desk=5204
20 sys desk
```

MENU "Menu Header" prints the heading for the pull-down menus at the top of the screen. The heading may be up to 38 characters long, for example:

```
10 menu=5234:scnclr
20 sys menu "Menu0 Menu1 Menu2 Menu3"
```

Store & Fetch respectively store the current screen in RAM, and bring it back again. They do not require any parameters and you can only have one screen stored at a time. For example:

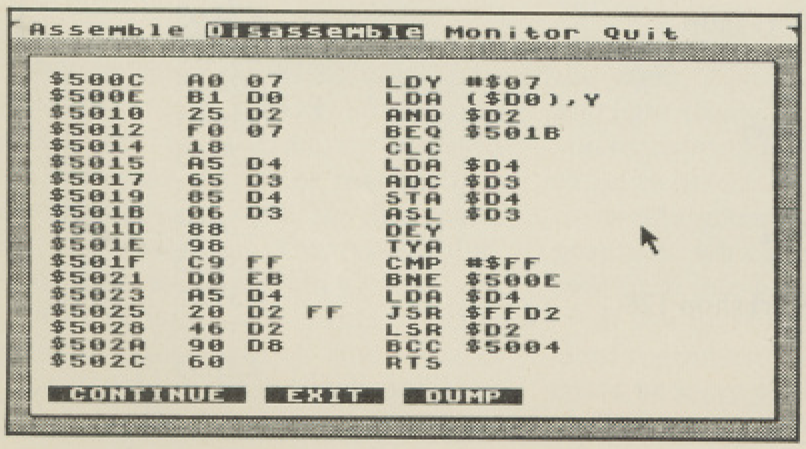
```
10 sre=5281: fetch=5316: desk=5204
20 key:sys sre
30 sys desk:char1,1,1, "Press any key to see keys."
40 getkey k$;sys fetch
```

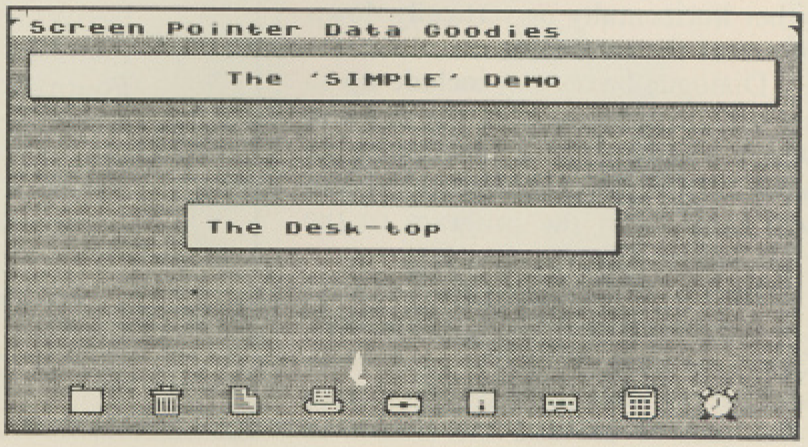
ICON,x,y,i will print an icon on the screen at any text position. 'x' is the x-coordinate, 'y' is the y-coordinate, and 'i' is the icon number. The ranges for these parameters are as follows:

-1 < x < 39 – Range of X-Coordinate.
-1 < y < 24 – Range of Y-Coordinate.
-1 < i < 15 – Range of icon number.

For example:

```
10 icn=5334:desk=5204
20 sys desk
30 x=int(rnd(1)*39) :y=int(rnd(1)*24)
:i=int(rnd(1)*15)
```





```
40 sys icn,x,y,i
50 goto 30
```

WINDOW,x,y,w,h requires four parameters and will set up a window on the screen, draw a border, and clear the window. 'x' and 'y' are the co-ordinates of the top-left corner of the window, 'w' is the width of the window, and 'h' is the height of the window. The ranges for these are as follows:

-1 < x < 38 – Range of X-Coordinate.
-1 < y < 23 – Range of Y-Coordinate.
0 < w < 39-x – Range of Width.
0 < h < 24-y – Range of Height.

For example:

```
10 window=5416:desk=5204
20 sys desk
30 sys window,5,5,10,10: print "Window  # 1"
40 sys window,10,10,10,10: print "Window  #2"
```

POSITION,x,y simply sets the co-ordinates of the pointer to the inserted values. It is used in conjunction with the LOOP command to set the start position of the movement. 'x' and 'y' are the coordinates of the pointer and are in the range below:

-1 < x < 40 – Range of X-Coordinate.
-1 < y < 24 – Range of Y-Coordinate.

For example:

```
10 pstn=5591
20 input"Pointer Coordinates"; x,y
30 sys pstn,x,y
40 ?"The Pointer is at"x;y
```

SHOW & HIDE respectively print and erase the pointer. When SHOW is executed, the two characters under the pointer are saved before it is printed. HIDE then retrieves the characters and puts them back on the screen.
For example:

```
10 desk=5204: pstn=5591: show= 5618: hide= 5680
20 sys desk;sys pstn,20,12
30 char 1,1,1, "A Flashing pointer!"
40 sys show:gosub 60
50 sys hide:gosub 60:goto 40
60 for t=1 to 200:next t:return
```

SPEED,s – as all the routines are written in machine code, if there were not a delay loop in the routines to move the pointer etc... it would zip around the screen and be totally uncontrollable. Also, people have different reaction times, and may prefer to have the pointer moving at a slower speed than others. This command will determine how slow or fast the the pointer moves. 's' which is the speed, must be in the range below:

-1 < s < 256 - Higher the value, lower the speed.

For example:

```
10 desk=5204: pstn=5591: speed=5722: move=5729
20 scnclr:input "Pointer Speed(999 to end)";s
30 if s=999 then stop
40 sys speed,s:sys desk:sys pstn,20,12
50 sys move:goto 20
```

MOVE is the most important command in SIMPLE's vocabulary. It draws the pointer, and allows it to be moved around the screen until the select button is pressed without erasing everything it moves over. For an example of it in use, see SPEED.

MAKE "Option1/Option2/Option-3...", sets up a pull-down menu Pull-down menus have a fixed size of 8 by 9 characters with up to eight options, and the last option is always 'Exit'. Each line of option text must be separated by a '/' character. 'm' is the menu number and 't' is the x-co-ordinate at which you wish the menu to be pulled down.

"01/02/03.." 72 Characters – Option text length.
-1 < m < 4 – Menu number.
-1 < t < 31 – Tab position.

For example:

```
10 make=5837
20 sys make "These/are the/options/for/menu/number/nought" ,0,1
30 print"Type m 1837" '
40 monitor
```

PULL,m pulls-down menu number 'm' and allows you to select an option by moving the reverse-video bar up and down, then pressing the select button. The parameter range is shown below:

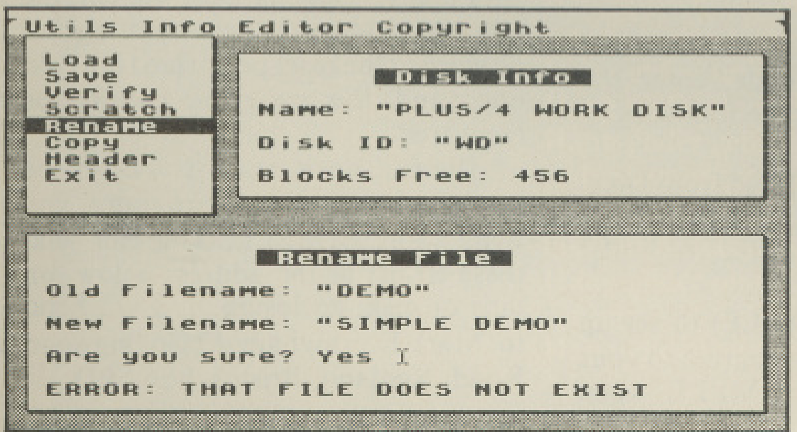
-1 < m < 4 – Range of Menu number.

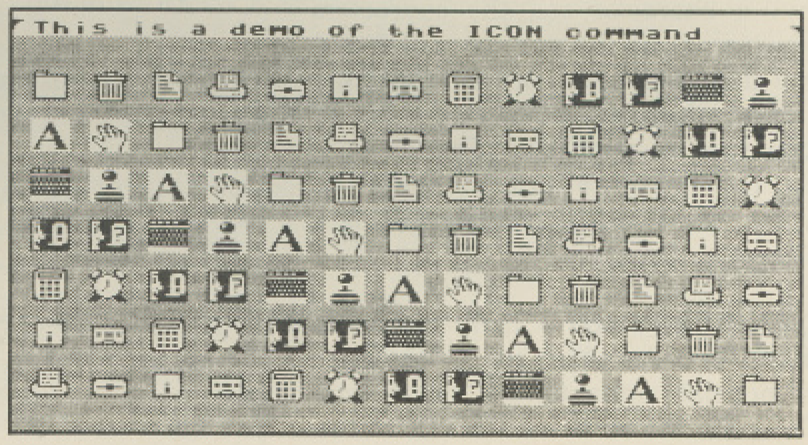
For example:

```
10 desk=5203: menu=5234:
make=5837: move=5729: pull=5937:
hide=5680: sre=5281: fetch=5316
20 sys desk: sys menu "0 1 2 3"
30 sys make "Menu 0/Opt2/Opt3/Opt4/Opt5/Opt6/Opt7/Opt8",0,1
40 sys make "Menu 1/Opt2/Opt3/Opt4/Opt5/Opt6/Opt7/Opt8",1,3
50 sys make "Menu 2/Opt2/Opt3/Opt4/Opt5/Opt6/Opt7/Opt8",2,5
60 sys make "Menu 3/Opt2/Opt3/Opt4/Opt5/Opt6/Opt7/Opt8",3,7
70 sys move :x=peek(216) :y=peek(217)
80 sys hide:if y  0 then 70
90 m=peek (3072+x+y*40) -48:if m  3 then 70
100 sys sre:sys pull,m:sys fetch:goto 70
```

In Use

ENABLE/DISABLE respectively enable, and disable the interrupt which keeps the character-set pointers





Screen-Code	Function
64	Desk Pattern
91-92	Pointer on a plain background
93-94	Pointer on a desk background
95	The I-Bar (Text pointer)
96-97	Border characters for menu heading
98-105	Surround characters for windows

Figure 1

pointing to the SIMPLE font. This is so that when you make a mistake in a program, the screen doesn't clear to a interference-type mess!
For example:

```
10 sys 6485:end
20 sys 6498:end
```

Type RUN 10 then type some rubbish. The screen will blink, and your error will be displayed.
Type RUN 20 then type some rubbish. The screen will go crazy! Type RUN 10 to re-enable the interrupt.

Co-ordinate Sensing – after a MOVE or POSITION command, locations $D8 and $D9 hold the current coordinates of the pointer. Thus, they can also be changed by POKEing directly.

Option Sensing – after a PULL command, the Option that was selected is stored in $D9 – the Y-Coordinate of the pointer. This is so that the pointer remains in a logical place after pulling a menu down. Before moving the pointer again, you must either assign the option number to a variable or store it elsewhere in RAM.

The SIMPLE Font – to improve the Reverse Video, and create a space for icons, a new font is defined by SIMPLE starting at the address $2000. This font can be redefined in the usual way by using a Character Designer.

The Icons – to squeeze all fifteen icons into a character set, the characters which make up the icons are not stored in a strict sequential order – they skip around alphabets etc... For a list of screen codes which make up each icon, see the reference table (Figure 1). Again, these can be redefined using a Character Designer.

Special Characters – there are some character codes which are neither icons or normal alphanumeric characters. These are shown below:

Redefining the Keys – the listing presented here is configured for a joystick in port 1, using the fire-button as a select button. If you do not have a joystick, or would prefer to use the keyboard, you can change the keys quite easily. The third section of the reference table shows the addresses you need to change the values in, and the relevant values for a joystick in port 1, and the cursor keys + RETURN. However, you can use any keys by poking their ASCII codes into the locations shown.

Starting the System – to take up as little memory as possible, SIMPLE is installed at the bottom of memory. It operates only in text mode, and you must not attempt to enter a Hires mode as this will corrupt the program. As it is at the bottom of RAM, the start of program pointers must be set before entering the SIMPLE Loader program, or before writing your own SIMPLE programs. This is done as follows and is *very important:*

```
POKE 10240,0:POKE 44,40:NEW
```

This sets the pointers. Now, you can begin to type in the loader or begin to write your own programs.
When you have typed in and debugged the loader, RUN the program and follow the instructions to save the code.

MONITOR
S "simple",8,1000,2801 – Save To Disk.
S "simple",1,1000,2801 – Save to Tape.

X – Exit Monitor.

Then to reload the code, enter the above pokes and type the following:

LOAD "simple",8,1 – Load from Disk.
LOAD "simple",1,1 – Load from Tape.

Type SYS 6485 (ENABLE) to set up the interrupt and now program to your heart's content. It's SIMPLE when you know how!

Using the Demonstration Program

Before typing the demonstration program in, you must set the bottom of memory pointers as explained previously. The program starts by loading the code file "simple" which should already have been saved as explained before. Tape users must change line 10 to:

```
10 IF C=0 THEN C=1:LOAD "SIMPLE",1,1
```

Once the program has been run, you can move the pointer anywhere on the screen. Experiment with pressing the select button. You will find that everything on screen will give a reaction of some kind!

Clicking on one of the icons at the base of the screen will result in a small window to indicate what each icon represents. Clicking on the desk-top, title-window, or menu-header will have the same result.

The four pull-down menus are Screen, Pointer, Data, and Goodies. The first three consist of lists of the SIMPLE routines. Selecting any of these will show a large window telling you the syntax, etc of the routine. These windows are closed by clicking on the reverse-video button saying "OK".

The fourth pull-down menu has four options: Input, Speed, Info, and Quit. By selecting Input, you can choose whether to use Joystick or Keyboard. This is done by clicking on the relevant icon.

When you select Speed, a window will appear in the centre of the screen with a bar showing the present speed. You can change this setting by pointing at the position on the line corresponding to the required speed.

Selecting Info just shows a window of general information about the program, which is removed by clicking on OK in the usual way.

Finally, selecting Quit will exit the program and reset the computer, but *be warned* the reset performed is a cold one, and the program cannot be recovered.

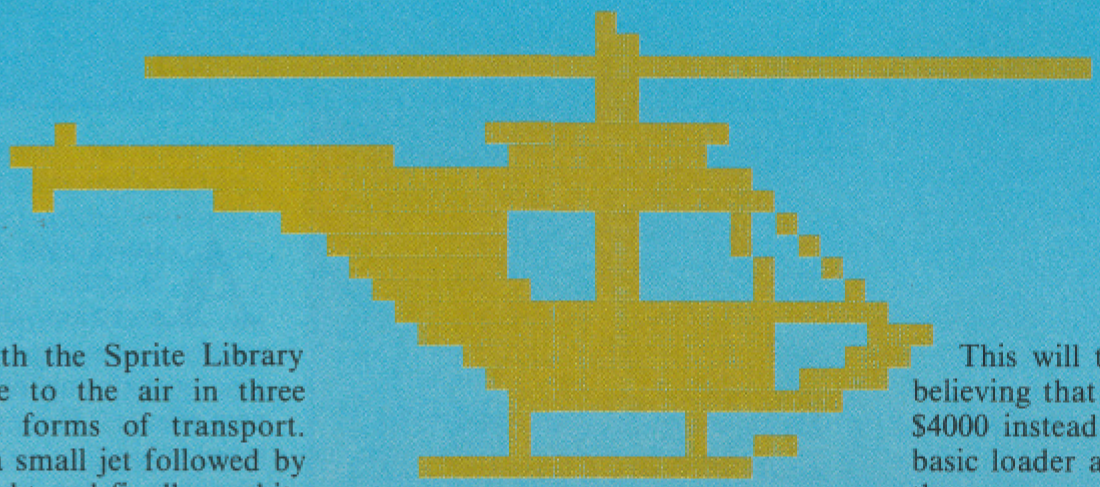
Please note that I would be interested to see any programs you write using SIMPLE. You can send these to me at the address below, on tape or disk, enclosing an SAE. Send to: Mark Everingham, 17 Collingwood Road, Redland, Bristol, BS6 6PD.

See listings on page 73

Sprite Library

This month our Sprite Library series takes to the sky with jets, helicopters and hang-gliders

By Mike Benn



This month the Sprite Library will take to the air in three different forms of transport. Starting with a small jet followed by a helicopter flight and finally a white knuckle ride under a hang-glider. Both the plane and the helicopter approach the viewer from the distance and turn to the right. The hang-glider is not animate as space did not permit. However, there should be enough key frame for you to fill in the missing animation.

All the sprite definitions require two sprites.

Getting it all in

Type in the basic loader as published and SAVE IT-DON'T RUN IT or it will self destruct. Before running the loader program you will need to reset the computer and type the following: POKE43,0:POKE44,64:POKE 16384,0: NEW and press return.

This will trick the computer into believing that the basic now starts at $4000 instead of $0801. Load in the basic loader and run it; if error free, the program will automatically save itself as a block of data. If you reload that data in the future remember to add a 1 after the device number. The data is saved in the following location $2800-$37FF.

The sprites run from 160 to 223 in a compromise to avoid the area $2000 traditionally set aside for redefined character graphics and to avoid the need of typing in line after line of data.

If only one or two sprites are required then use the following formula: (Sprite block No.-160) *40 + 190 = the date line number at which that sprite blocks data starts. Remember to type in the following three lines of data and alter the variable BL to the number of data lines you have in your finished program, less 1.

The small basic program AIRCRAFT DISPLAY will variably animate the sprites in both non-expanded forms on the screen simultaneously. To hold a sprite enter the same number for Start and End.

Any Sprite Editor program will enable you to change and adapt the individual sprites to your own requirements.

See listings on page 73

HEX	DECIMAL	DESCRIPTION
A0-B7/	160-183	Jet plane approaching and turning to the right
B8-CA/	184-202	Helicopter approaching and turning to the right
C0-C3/	192-195	Helicopter blades turning while helicopter hovers facing the viewer
CA-CD/	202-205	Helicopter blades turning while helicopter hovers facing right
CE-D5/	206-213	Hang-glider turning to the right
D6-D7/	214-215	Hang-glider flying upwards to the right
D8-D9/	216-217	Hang-glider getting ready to land or has just taken off
DA-DB/	218-219	Hang-glider on the ground just about to take off or has just landed
DC-DD/	220-221	Hang-glider on the ground, pilot under canopy
DE-DF/	222-223	Hang-glider on the ground

Games Update

Well, as to be expected all is quiet on the games front this month. Few releases are around and we all eagerly await the build up to the Autumn releases



Questron II

A quick visit to my local W.H. Smiths reveals that their software shelves have large numbers of compilations and sequels among their top titles, with very few new releases. Hopefully, Autumn will bring a host of exciting products in what has, so far, been a disappointing year as far as games are concerned.

One new feature this month is the introduction of a roundup of Amiga games. The sixteen bit invasion marches ever on.

Commodore 64

Strategy is very much the flavour of the month with one fantasy role playing game, two wargames and one classic board game comprising the main offerings.

Questron II (US Gold/SSI) sees you trying to destroy the Book of Evil. The only way to achieve this is to travel back in time and prevent it from ever being created in the first place! This involves you seeking out six mad sorcerors and your journey will take you over two continents, through dungeons, catacombs and castles before you achieve your goal.

The game is entirely menu controlled via either joystick or keyboard although I found the keyboard much easier. Estimated playing time is thirty to sixty hours. This is one of the simpler fantasy role playing games currently available and

would probably appeal more to the novice or younger player of this type of game.

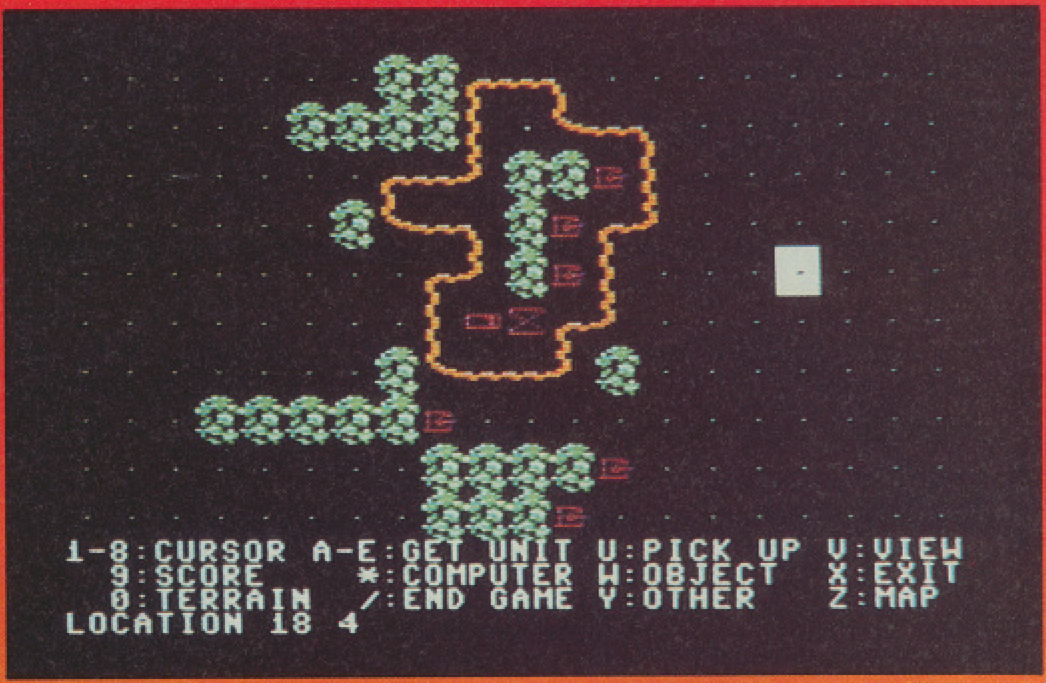
Two wargames have also been released from the US Gold/SSI stable. *Panzer Strike* cannot claim to be short of ambition, covering as it does, the entire Eastern Front campaign, the Western Front in 1940 and the North African campaign. Should that be insufficient for your needs, you can always design your own battles and campaigns. Despite the size, symbols still represent individual tanks and to add extra realism, armour ratings on these vehicles have been segmented into the front and side of the hull and turret and top. Every conceivable ground weapon has been included in this simulation from artillery to trucks, mortars to tanks. Not surprisingly, this game is recommended for advanced players only!

Somewhat easier is *Sons of Liberty*, which contains introductory, intermediate and advanced scenarios. Set in the War of Independance or, as the Yanks like to call it, the Revolutionary War, you get the chance to re-enact the three major battles – Bunker Hill, Saratoga and Monmouth.

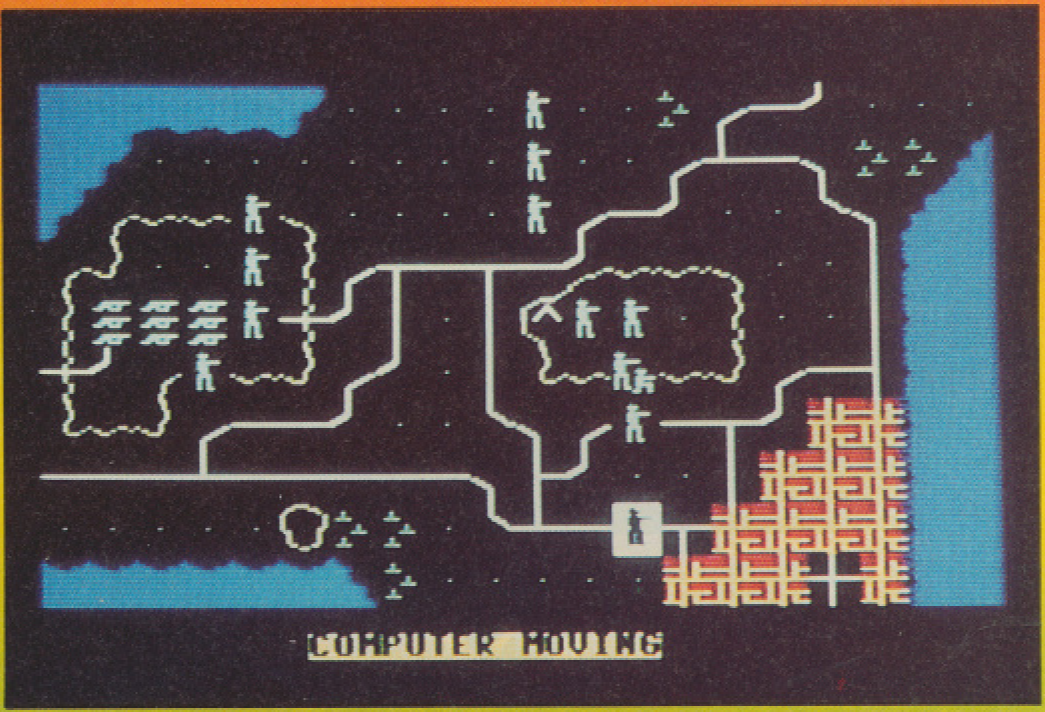
Bunker Hill was a relatively small conflict. Only a few thousand men fighting over a small area. Not only did the Brits come second at Saratoga, but the major defeat caused the French (boo, hiss) to decide that this might be the perfect opportunity for them to also declare war upon us so that we now had to fight on two fronts. The battle of Monmouth was the largest of the three battles and the time when George Washington really came into his own as a leader of men. So here is your opportunity to rewrite the history books and ensure that the good old US of A really should be coloured red on all the maps!

From Leisure Genius comes that old favourite, *Monopoly*. This game has probably caused more inter-family rows than any other in history. Monopoly players seem to come in two forms. Those that play to win at any cost and those who don't. The two factions are totally incompatible with neither side being able to see the point of view of the other.

The computer version allows you to play against either human or computer opponents or any mix of the two. You can opt for a long or short game and games can be saved midway through. Control has been made as



Panzer Strike



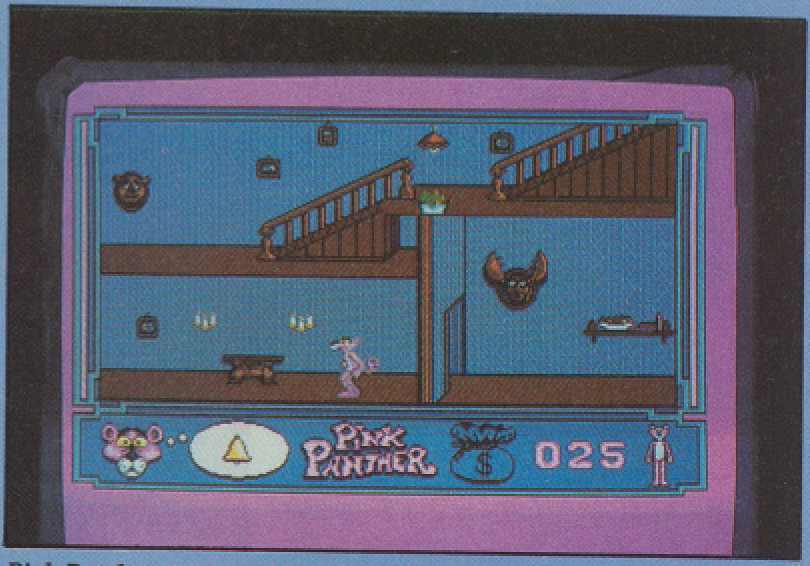
Sons of Liberty



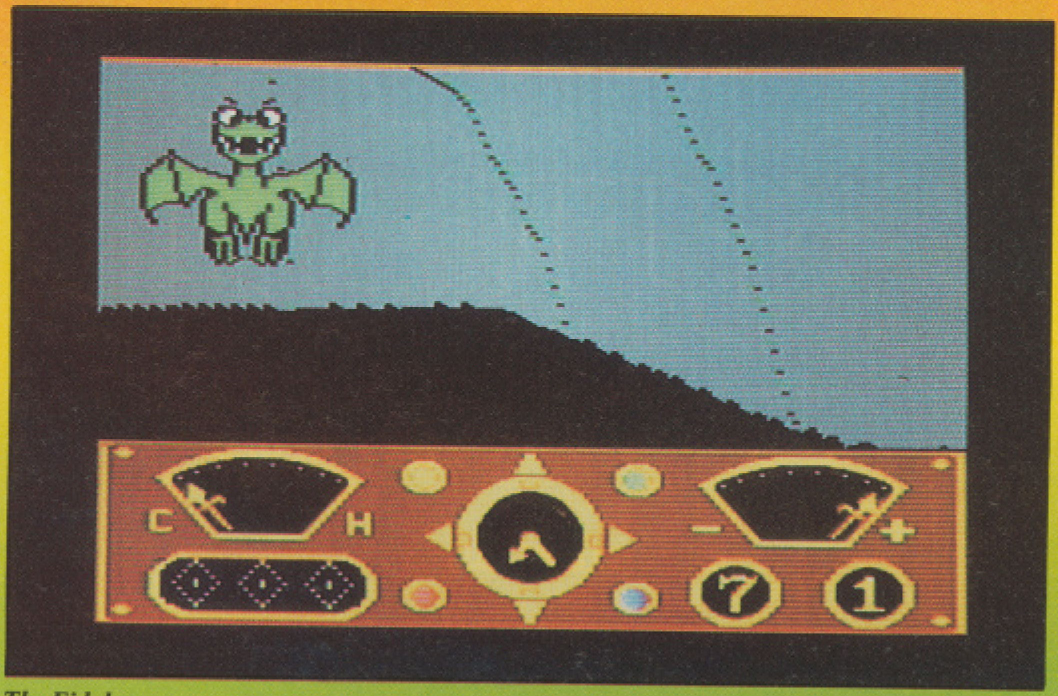
Monopoly



Seven Cities of Gold



Pink Panther



The Eidolcn



Bruce Lee

user friendly as possible and all the features of the original game – mortgaging, auctioning and going to jail have been included. My only disappointment is that with the computer looking after the money, it is no longer possible to cheat! (No prizes for guessing to which faction you belong – Ed.)

Few budget games are noteworthy this month. On Mastertronic's Ricochet label comes *The Eidolon* a 3-D maze game featuring some excellent graphics for the monsters. You need to find and collect jewels and fireballs before being allowed past the dragon and onto the next level.

On the Americana label comes *Bruce Lee* a platform game with martial art elements. Bruce has to collect all the lanterns in an area before a door opens allowing access to the next level. At the same time, he must avoid the ministrations of ninjas and the Green Yamo. The game looks very dated but is still good fun to play.

Four games for under a pound each is the boast of Microvalue's third compilation. The games are *Equinox,* an arcade adventure, *Darksyde,* a shoot 'em-up, the Big KO, a boxing simulation and a platform game, *Mouse Trap.*

Finally, with timing that is about as good as one of Vinny Jones' tackles comes *FA Cup Football* also on the Ricochet label which arrived in the office two days after Wimbledon had fluked their win over Liverpool. As a game, it is about as interesting as Don Howe's tactics which, for those of you who care little for football is not very much.

Amiga

From those masters of the adventure game, Infocom, comes *Sherlock* in which eponymous hero must once again outwit the evil professor Moriarty. There is a twist to the plot though. Because Moriarty knows how Holmes will react and can lay traps accordingly, Holmes decides that it is you, Watson that must solve the crime. The Crown Jewels have been stolen and if they are not recovered within 48 hours, then the theft will be exposed and the government disgraced. Sherlock is available from Activision and is highly recommended.

Exploration is the name of the game in *Seven Cities of Gold* from Electronic Arts. Starting in 1492, you must discover and explore as much of the New World as you can. Sail up rivers, find Inca temples and gold mines. You must decide whether to trade or fight in order to persuade the natives to part with their gold. Supplies of food, men, goods and treasure all have to be carefully managed as your ships can only carry so much. I first saw this game a few years ago on the C64 and enjoyable though it is, I'm afraid that little attempt has been made to use the graphic and sound capabilities of the Amiga to the full.

Following on from their successful Barbarian, Psygnosis have released *Obliterator* which has the same sort of icon controlled gameplay but is in a futuristic setting. As the last of the Obliterators, you are beamed aboard an alien ship with the task of finding some vital computer datapacks before destroying the ship and thus saving the Federation. You only have limited weaponry to start with but more can be picked up as you go.

The Three Stooges is the fifth Cinemaware title to be released by Mirrorsoft. Curly Larry and Moe are involved in trying to save an orphanage from the evil Mr Fleecem. Raising the money to pay off the mortgage involves a series of subgames and gives you the chance to get involved in custard pie fights, trying to eat crackers before a vicious oyster does and having dodgem car races when trying to reach the operating theatre. As with all Cinemaware products, presentation is excellent. Gameplay is limited though and some may baulk at paying such a high price for not very much game.

Fans of the films and cartoons will no doubt be eager to have a look at *Pink Panther* (Magic Bytes/Gremlin) but I fear that they might well be disappointed. Our hero is broke and takes up various jobs in order that he might burgle the properties. Unfortunately, all his employers turn out to be sleepwalkers and PP has to stop them from bumping into things as well as stealing whatever he can get his hands on. As well as pushing the boss in a safe direction, PP can also leave objects lying around to help him such as catapults and see-saws. These might also prove useful when trying to avoid the unwanted attentions of Inspector Clouseau. Although the game is very well animated, the gameplay is incredibly difficult to get into and it was a case of frustration rather than anything else that stopped me going back for any more. YC

Making Music

Continuing our music series, this month we look at the use of interrupts when playing background music

By Peter Gerrard

There are many ways of playing background music using interrupts, and in the next few pages we'll be looking at just two of them. Both use the same kind of technique, so we'll see how that works before getting down to the serious business.

When playing a tune in Basic, the information is read as a series of data statements and POKEd into the appropriate registers. In machine code we will have to have the music stored as a collection of numbers in memory, and use strange incantations and terms like 'load the accumulator' and 'store the accumulator' in order to make the appropriate noises. To those of you who are frightened by machine code programs, 'load the accumulator' and 'store the accumulator' can be thought of, at least as far as this program is concerned, as being analagous to something like A=PEEK(register) and POKE(register), A respectively. In other words we're going to be looking at a collection of memory locations, getting information from them, and then storing that information in the correct registers to make a sound.

As in the Basic example we will need a dummy set of data, or dummy information in memory, to tell the program that it has reached the end of the current data for the particular tune that we've set it playing, and since storing a -1 in a memory location is rather difficult (you try POKE 832,-1 and see what happens!) we'll be using the value 255, since a high frequency value of 255 makes little sense to the SID chip and simply produces an extremely high pitched whine. If you really want to make a high pitched whine, then use value 254 instead and leave the program to finish of its own accord.

Using Voices

In the program that follows we will be using voices two and three to play the background, interrupt driven music. Voice one will be left free for other devices and, as you might readily appreciate, voices two and three can also be used for miscellaneous noises and sound effects, since an interrupt driven tune, once started, cannot be diverted unless we tell it to be diverted. That is, voice three (in our example anyway) is being used to play a sequence of notes, and it will play those notes over and over again regardless of anything else we might tell it to do.

The waveform is stored in memory by the interrupt routine, the values for the high and low value frequencies are read in by the interrupt routine, and so anything else that we might do outside the routine is swiftly overridden by the routine itself. We might decide to make a noise using voice three set to the white noise waveform, but our machine code program soon recovers from this and carries on without skipping a note. So, we can play our two voice tune, have one voice left over to do whatever we want, and *still* have voices two and three to use if we really want to.

A second important fact to remember about the interrupt routine is that it can be played at a varying rate. I once heard someone else's music program, and was quite impressed by the way in which one could speed up, or slow down, the rate at which a background tune was being played. It was only in devising this program that I realised how trivial an exercise that was!

Consider the following – the interrupts on the Commodore 64 are serviced every fiftieth of a second or so, and if we're playing an interrupt driven tune, this means that fifty times a second the routine is going to be activated and a note (or two, since we're using two voices) will be played. This is obviously much too fast, since few of us are capable of registering that many notes per second. We're not all fans of The Ramones you know!

So, we insert a delay loop so that a note is played only every (say) tenth interrupt. This not only slows the rate of note playing down to an acceptable level, but it also allows other things to be serviced by interrupts as well. For example if we wanted to we could have another interrupt routine in memory that was moving sprites around, or rotating user-defined graphics a la Tony Crowther!

Such a routine might look like this:

```
C000 LDX $CFFF
C003 INX
C004 STX $CFFF
C007 CPX # $0A
C009 BNE $C012
C00A LDX # $00
C00C STX $CFFF
C00F JMP $(routine)
C012 JMP $EA31
```

Assume that we have diverted the Hardware Interrupt Vector so that it leaps off to $C000 instead of $EA31 as usual. Then, everytime it gets there we load the X register with the content of memory location $CFFF (X=PEEK(53247) if you like). Then we increment the X register (X=X+1) and store the new value at $CFFF again (POKE 53247,X). We then compare this new value with $OA (Does X equal 10?), and if it is not equal then we jump to location $C012 (if X< >10 then go

to location $C012). If this is the case then program execution simply jumps off to the normal interrupt routine at $EA31.

However, on the tenth time we get here X will have been increased so that it *does* equal 10, and so instead of jumping to $EA31 straightaway we must first of all reset our counter by loading the X register with zero and storing it at location $CFFF (X=0:POKE 53247,0) before jumping off to our own routine, wherever that might be stored in memory.

Thus we can increase or decrease the speed at which an interrupt driven routine is serviced by just altering the value that we're comparing the X register to. At the moment it's set to ten, so if we lower that to five then the tune will be played twice as fast. Increase it to 20, and everything goes at half rate. Quite easy, really.

Before presenting the routine itself, we just need to explain now how it works. Not by going through it line by line, as in the example above. If you can follow machine code programs then you'll be able to do it for yourself, but if you can't then this is neither the time nor the place!

How It Works

Voice three is used to repeat a tune over and over again. The information for this tune, in low value/high value order, is stored in memory from $9E00 to $9EFF, and so we can have something like 128 notes in this particular little riff.

Voice two is used for the much longer tune, and can indeed be said to represent our musical soundtrack. This one is stored in memory from $9000 to $9DFF, in low value/high value order again, thus giving us the space to store some 1920 notes. Quite a lot, and the top of end of Basic memory is always a fairly safe place to save information. Later on we'll see how we can do away with even this, and still have space for a long soundtrack.

There are limitations however with the space for all these notes. Voice three is continually playing a riff; ideally one that is linked in to voice two somehow, and if we have (say) 96 notes in the riff then it makes sense for voice two to be playing a tune that has a multiple of 96 notes in it, otherwise things soon begin to sound horribly discordant.

How do the notes get there in the first place? It is an easy task to modify the synthesiser program so that, as well as playing the notes, the low value and high value frequencies are stored in memory, from $9000 or $9FFF depending on which voice you want to have saved. Include the option to delay a note or introduce a pause (the space bar, for example, could be used to put the values of 00,00 in for low value/high value, which is as effective a way of introducing a one note pause as any), and there you have it. Regard it as an exercise in programming the SID chip and getting the brain cells working, and if enough people write in and complain perhaps the editor will give me space to expand on the topic!

Having got the tunes in memory, we must then load and run the program called Interrupt, and music will be the result. For those of you not of a musical turn of phrase (like me) then the final program in this section will put a simple boogie riff into memory for you and allow you to listen to it. Not very musical, perhaps, but everyone has to start somewhere! YC

See listings on page 73

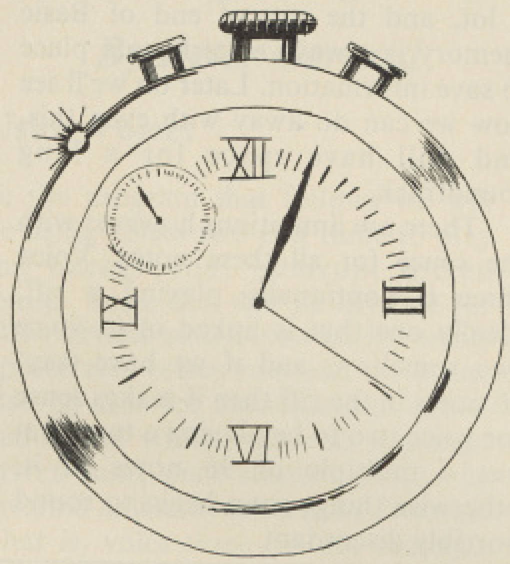
A Short Interlude

Overcome the problems of using more than one interrupt routine with this handy utility

By Michael Tinker

The 6510 processor used in the Commodore 64, like other processors is able to handle interrupts. The Commodore 64 makes the use of interrupts particularly easy by placing the ROM interrupt handling routine vector in RAM. This is located at $0314/$0315 (lo/hi format).

The interrupts take place 60 times per second and are a powerful programming tool. There are many short programs which use interrupts, available in printed form, from books and magazines.



Using Several Interrupts

The aim of this short machine code program is to make the use of several interrupt routines a lot easier. (I developed the program initially for a game I was writing which required several interrupt routines to operate at once.) The routine is however very useful for utility routines. First let us examine what the problems are in using more than one interrupt routine.

Each program is usually published as an individual, stand alone program which will usually have the following format:

```
SEI          !set interrupt disable
LDA #$00
STA $0314    !place address of new
LDA #$C0     !routine into interrupt
STA $0315    !vector
CLI          !clear interrupt disable
RTS

*=$C000      !start of routine
...
...          !main interrupt routine
...
JMP $EA31    !exit to ROM interrupt
              routine
```

This format means that two interrupt routines cannot be used together unless they are modified by changing the JMP $EA31 at the end of the first routine to point to the second. The start routine to change the interrupt vector at the beginning of the second routine then becomes superfluous.

The short length of most interrupt routines will allow more than one to be in operation without any noticable reduction in the computers speed, and before clashing with the next interrupt. Sixty interrupts per second, assuming an average of three clock cycles per instruction, allows about 5,000 instruction to be carried out before the non-interrupt program grinds to a halt.

The solution I have come up with is to keep a table of the interrupt routine start address in the form of a small array similar to the BASIC DIM A(5). The control program is called on each interrupt, and in turn will call each of the routines which have their address in the table.

To use this method, only small adjustments to the interrupt routines are required. First of all remove the usual start routine which usually changes the interrupt vector, as this task is carried out by the control program. Secondly, change the JMP $EA31 at the end of the routine to RTS. This is because the routine is called by the controller with a JSR instruction.

· The routines can be added into the table in any order; they are each called in turn starting with routine 0 at the start of the table.
(N.B. ensure that the interrupt vector is not pointing to the controller when you change the table, as it will attempt to call a routine which is not present, if the address is only partly changed when there is an interrupt.)

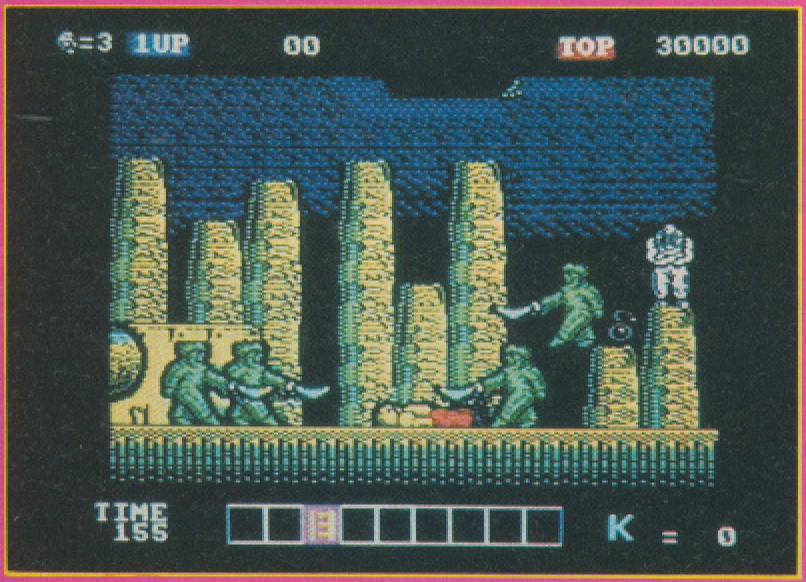
The controller is particularly easy to use because there is no need to predefine how many routines you wish to use, entries in the table of $0000 will not be called, therefore if you wish you can even stop all the routine calls. This is particularly useful when starting to use the computer when you may not know how many routines will be used.

Getting it all in

If you enter the Mikro Assembler listing first save a copy incase there has been a catastrophic error and then assemble; a SYS call to 49152 will start the program. The basic listing is easier to use as there is a checksum error check as part of the listing. Also when the program is run, the cursor will reappear over the command SYS49152 after a short delay, so all that is required is a further press of the return key and it is away!

Next month we will look at a program of more general use which will have commands to add and remove routines from the table and also list the contents, showing the addresses which are being called. Until then, happy interludes!

See listings on page 73



Step out on the Russian steps with a fire-breathing circus strongman

Have you ever been killed by a skeleton riding an ostrich? It's pretty safe to bet the answer will be no! In fact there's only a minimal risk of this occuring to the likes of you and me at the time of writing. Karnov however, an Russian ex-circus strongman, appears to have an affinity for this and other bizzare disasters, guaranteed to occur as frequently as wet afternoons in August! The game tends to remind me of wet afternoons too, nine levels of the most pathetic graphics you can imagine – more later!

The game opens as Karnov is zapped into existence by a bolt of lightning. Your task is then to battle your way past marauding creatures of every description, and a few that even defy description, collecting useful objects along the way. At the end of each level, you'll need to load the next part from tape – an annoying little distraction. The jist of the plot is as follows: the Russian village of Creamina (Karnov's retirement home) has a secret – it's the hiding place for one of the world's most awesome treasures (no, not Paul Daniel's wage packet), stolen by Ryu, an evil wizard, who left behind a few monsters to punish the villagers for hiding the treasure in the first place! Karnov naturally decides he's the man for the job (obviously after the insurance reward), and sets off to find the pieces of the map that will lead him to Ryu, the treasure and the game's end.

Now to the graphics. The background features are solid but colourful while the moving objects (actually hardware sprites) are well detailed. Where the whole thing falls to pieces is the animation. All the sprites are animated using just two positions, giving them a very jerky disposition, but worst of all is the big blank space behind them! OK, if this was a Plus/4 or Spectrum game, this is to be expected, but there's no excuse for scrappy graphics on the C64!

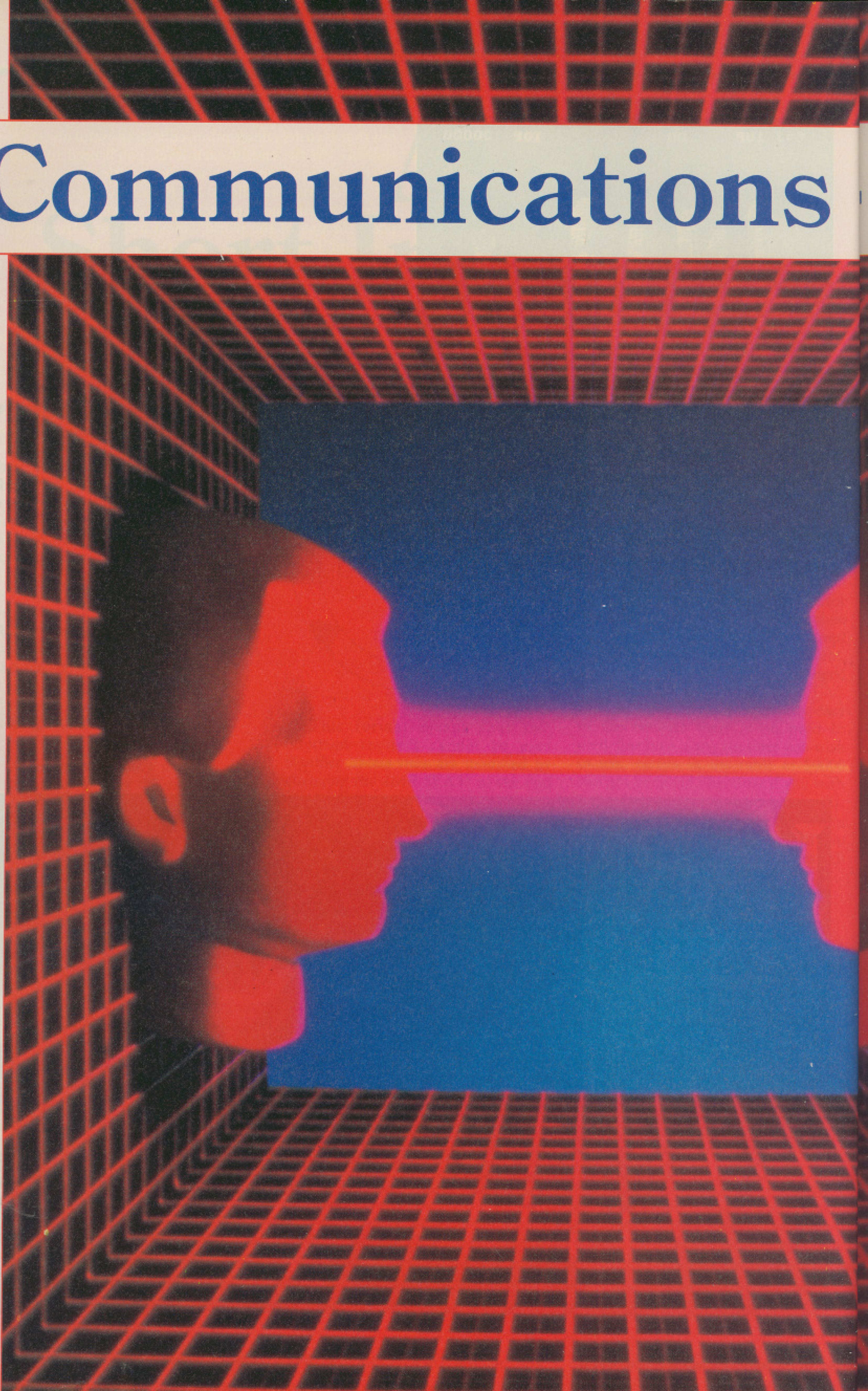
As for the sound, equally pathetic – not even worthy of a VIC 20! Assorted clicks and squeaks with a brief snatch of monophonic Bach each time you die. Having said all that, I did find the game challenging and mildly addictive, although not addictive enough to persuade me to buy it!

F.R.

Touchline:

Title: *Karnov.* ***Supplier:*** *Electric Dreams Software, Terminus House, Terminus Terrace, Southampton, Hampshire SO1 1FE.* ***Tel:*** *0703 229694.* ***Machine:*** *C64/128.* ***Price:*** *£9.99 (Ca) £14.99 (Disk).*

Communications





s - the way forward

We look at the history of computer communications, where it came from, where it is now, and where it is going

By Mary Branscombe

The essential techniques of sending data down a telephone line has a history going back 160 years, and some of the common jargon terms that have evolved alongside the more traditional computer buzz words can be traced right back to the days of the Telegraphic transmission systems in the Victorian era!

The earliest form of telegraphy – being the first form of totally electrical and "wireless" communications systems, and probably the earliest form of electrical message sending, was the remote actuation of electric relays to punch holes in a piece of paper. The letters of the alphabet were defined by a stream of these dots each representing a 0 or a 1 and were thus defined between a mark and a space signal.

This goes back as far as 1840, and the first reliable machines to do this were traditionally kept for military purposes, although the government found that the new devices were very useful for transmitting information from one part of the country to another.

The mark and space have been replaced by holes punched into larger paper tape, larger holes for mark, and smaller ones for space – this code is called Baudot, and is in common use today in some of the more backward corners of this planet. After this development, longer distance communications was made possible by synchronising the sending and receiving stations by beginning each letter with a start bit and concluding it with a stop bit, so over the years the standard was to send seven bits of data; the start bit, the stop bit, the code for the letter and, of course the mark.

When after 100 years or so – technology was increasing dramatically – there was a need to access computers remotely. Telegraphy was the obvious way of doing this as the technology could easily be interfaced with the new computer systems and the information was readily translatable. But back in the 1950s computers were not the machine we know today, processing was never expected to come up with instant results; jobs were assembled in batches (hence the term batch processing), fed into the computer by paper tape and then run. Instant collection and processing of data was then considered quite amazing!

So the first major application which computer communications had was to ensure that computers were fed with up to date data, the paper tape was then quite a fast way of inputting large amounts of data and indeed this saved a lot of time with computer entry staff, who were employed exclusively

to enter data as quickly and as accurately as possible.

Barbaric wasn't it?

Typical communications speeds were at 50 or 75 bits per second and this is NOT the baud rate of a communications system. Baud rate and bits per second are more or less identical at lower communications speeds, but at higher rates bits are transmitted in different ways other than changing the state of a signal from high to low (and vice versa), additional information is transmitted by detecting the phase state of a signal, so 1200 bits per second full duplex is actually achieved by a 600 baud signal using four phase angles (two for each end of the communications stream).

In days gone by there were no friendly on-line systems, no real remote access, terminals were still yet to have screens (let alone faster storage devices) and there were literally thousands of protocols, variants and alternative communications systems cropping up all over the shop.

The Telephone Network

By this time, all sorts of changes were taking place. The Telex and Telegraphy network, originally set up to handle data, were taken over by voice grade circuits (the earliest was Bell's telephone design from 1876). And likewise, communications systems were being adapted to make the most of the telephone system by using audio tones to represent the state of high and low in a pair of twisted wires. Initially this was merely one directional transmission and receiving (half duplex as the terminology had it), but by cleverly using two different signals, full duplex (or two way) communication was possible.

Improved digitising and sampling systems after the war (primarily) developed for the encryption systems set up for England and America to communicate via the war) led to faster baud rates – although 50/75 bits/s were possible. The next step was to move to 110bits/s, this soon moved up to 300 bits/s and only recently has the limit been reached for accurate data transmission along a voice grade telephone line – 2400 bits/s.

Over the years little has actually happened to the way the data is transmitted along a line; the start and stop bits of the baudot code have been retained, but instead of a simple 5 bit signal used to transmit the letters of the alphabet, a new system has been employed whereby a 7 bit extended code can be used to transmit the letters of the alphabet. This in fact was as a direct need to transmit data other than letters – 128 more in fact.

To reduce the number of errors in transmission due to line noises (remember we are still talking about voice grade telephone lines – and think how noisy they can be!), a further bit was added to the stream of data – the parity bit, which adds up all of the bits in the main characters and then, depending on whether the result is odd or even, generate a 0 or a 1.

This system gave birth to the world's most adhered to standard – ASCII, the common name for the alphabet – naturally IBM came along with their own variant, but as the PCs use ASCII instead of their own EBCIDIC system (used on mainframes). I think you will agree that IBM are licking their wounds and trying to keep quiet about this one!

Higher Speeds

The fastest speed possible on an ordinary voice grade telephone line is usually regarded as 1200 bits/s. Beyond this, noise on the line, due to "noisy" exchanges and poor cabling has made anything faster very difficult. 2400 bits/s is becoming more common, and indeed the standard transmission speed of TELETEX services, the high speed version of telex.

Services

Communications these days is easy for the microcomputer user – most computers have a common facility built into the hardware, be it by using the standard RS232 interfaces, or by some such similar serial communications.

Software too abounds for many computers. This is a far cry from the first days of computer communications, when the required "software" for communicating with another computer involved making direct links to a processor's data bus, and then fooling the computer into thinking it was reading from a very fast keyboard!

These days communications packages are very sophisticated and can handle many different baud rates. They can operate in either full or half duplex modes; they can effectively emulate many different types of terminal, from DEC's VT52 and VT100 and the Hazeltine to the more modern graphics terminals and professional workstations that support graphics.

Software is fast becoming more sophisticated than the modems that are used to communicate with the telephone line, and the faster they are, the more sophisticated the software needs to be. For example, the new generation of 2400 bits/s modems can now support special hardware encryption devices that ensure your data is unreadable as it goes down the line. The only way you are to retrieve this data is to use a similar modem with the same decryption device at the other end.

Communicating with a micro has become more popular than with, say, a data terminal. The reason is that a computer can be programmed in software which would cost many hundreds of pounds to achieve in hardware! Many of the initial sales of Amstrad PCs was not to the business sector, but to those requiring cheap and reliable terminals – a VT100 terminal can cost anything up to $1000 – but with a piece of £25 software running on a £400 machine, you have a terminal *and* a computer with on-line storage, plenty of buffer memory with all the extras attached.

Communications has really come home for the micro.

Bulletin Boards

For the micro user, a service has sprung up since the mid-1970's called the Bulletin Board is the ideal place equivalent of the bulletin boards you may have at work or in a social club.

Hackers are remarkably solitary creatures, only mixing with their own type –where possible, and to this end, the bulletin board is the ideal place to exchange ideas and programming hints and tips, so it is small wonder why these services haven't taken off and spread all over the world.

On a Bulletin Board, you are likely to find a great many messages –have a browse through a few and then you will see that a great many of them are requesting specialised information about the inner workings of a computer of a compiler. Occasionally

you will find a message saying something to the effect that they will be starting a BBS (Bulletin Board Service) of their own, so why not ring them up on their number with their modem set to such and such and have a chat!

I am an owner of a small number of well used computers – all of which are hooked up to modems at some time or another. I ring up BBS's daily, and this is why I run up £150 telephone bills a quarter! But the returns are much more valuable than £150. I have downloaded priceless public domain software instructions on how to build my own Winchester drive for my computer using surplus stocks (that one saved me £400!) and the interchange of information is friendly, informed, intelligent and varied.

Every now and then we receive calls from "hackers" asking for a few numbers to try out, and I often oblige with a few of the best known free services. These people often log on for a few minutes – have a look round and then disappear forever – this isn't hacking – and I would contest that breaking into government computer establishments is hacking – that is sheer vandalism (sorry, but my favourite BBS system was brought to its knees by a bunch of nerds). Real hacking is carrying on the information transfer and cameraderie among a close knit group of computer buffs. You don't have to be clever, but if you know something and you have something to say or offer, then welcome to the club!

It is true that BBSs have special user groups, or access to certain parts of the BBSs hard disk that are unobtainable to the usual people, and entry to these is honorary and entirely at the discretion of the Sysop (the manager of the BBS). I am a member of one and I wear it like a medal because it shows to each and every other member of the SUG (Special User Group) that we all have something special to offer each other, be it technical help, knowledge of a language, access to certain information regarding the BBS software (a great deal of which is maintained by ourselves and friends) or simply because you can offer some very useful information about improving services.

In short, accessing a BBS is like entering a group of friends that communicate all over the world – in much the same way that a pen pal does with paper, pen and ink!

Professional Services

For the businessman there are a great many services on offer. Financial data services are available, but they have to compete with the excellent services offered by Prestel and CEEFAX although specialist services offer less general information. Scientific databases offering access to files and programs all around the world are hooked up (in the UK at least) by JANET – the Joint Academic NETwork, and access to American computer systems at Berkley, MIT and others are possible by linking up JANET through PSS, British Telecom's Packet Switching System and the international version called IPSS.

Univesity computers are mainly about powerful processing, and it is often a good thing to try and gain access to these systems if you have particularly large applications to run on these machines. Or if you need to access specific languages or information services that support the science and engineering faculties around the world.

Polytechnic computers are probably the least interesting computer systems available to the traveller. They offer simple applications, but often offer a back door entry into university computers as they often have a PAD – very useful way of accessing other computers from a remote computer!

PADs are also very good at hiding your tracks if you are a hacker interested in prying into the inner workings of a computer system. This isn't merely done by logging on and typing in a few commands, but by reading up on the operating system manuals, getting information out of a library, and asking your friends on a BBS whether they have had any experiences of such and such.

The spirit of hacking is learning... so go on and learn something!

The Future

Not being particularly good at interpreting the future, I can only state what may happen and not what is going to happen. Computers are going to have to get faster and faster. It is true that even micros that move away from IBMs self-imposed exile, are using more and more powerful processors and have more and more memory that needs to be filled up more and more quickly. After all, if you have ever heard the old adage, "The program expands to fill the available space", then you will no doubt see the futility of accessing a database with a modem running at 300 baud! How are you going to upload 128K of program file using a slow speed?

The way around transmitting large amounts of data down the line is to go digital. By using a special digital data line, you can theoretically go as fast as 360,000 bits/s (that's a far cry from the 50 bits/s 160 years ago!) and even after speeds are possibly by splitting the telephone line so that the Rx line (the receiving line) is on one number, while the Tx line (transmitting) is on the other line!

Large mainframe establishments are using parallel communications over eight telephone lines nowadays, and with the data traffic being measured in gigabytes, the future of parallel communications has to come down to micro level in the next few years.

Digital traffic require special leased lines at the moment, but with the inclusion of X25 (which is just about working – snigger) this leads the way to parallel transmission down varying frequencies, so you could transmit a signal running on a carrier of 600 Hz and 1200 Hz simultaneously, running the Rx and the Tx signal at the same time down the same cable with no loss of data.

Light transfer is another option. Using fibre optical cable, modems may soon be just transmitting bursts of light down a cable straight into the telephone network. With this sort of technology, only the speed of the hardware is the limiting factor, so transmitting at 360,000 baud may just be a slow option –certainly we are aiming for data transfer rates that are faster than disk drives by the year 1990.

Of course this will all happen overnight! Data cables using fibre optics are already in wide use worldwide, but to be used over any great distance requires a revolution in manufacturing the optical cable, which is *very* expensive at the moment.

Thankfully the cable manufacturing companies are developing fibre optics so there will be competing interests artificially keeping the price of this new technology high.

Faster computers, faster communications lines and faster modems, when will it all end? Never, I hope! YC

Byting into the 6510

ROM routines can save you a lot of work and hassle. But first you'll have to learn how to use them – read on

By Burghard-Henry Lehmann

In my opinion the best way to learn machine code programming is to jump straight in the deep end, and start designing the routines yourself!

However it would be silly not to take advantage of the library routines already resident in your C64, namely the Kernal and Basic Interpreter routines. The saving in time and memory should be obvious. After all, why re-invent the wheel?

However, on the other side of the coin, the ROM routines in your C64 have been written specifically to operate the computer without it constantly crashing. Therefore a great number of safety checks have been built into these ROM routines. This has the drawback of making the routines slower and less efficient then those you design yourself.

However, if you ask yourself the question, "Is it really vital for the routine I need to execute that split second quicker?", then you can make a prudent decision, i.e. whether to write a routine yourself or use its slower counterpart already in ROM.

As a whole, ROM routines, when used correctly, execute quicker than Basic. And this is an advantage you should not forget!

The other disadvantage in using ROM routines is if you want to make your programs compatible with other computers. For example the BBC computer uses the 6502 processor which is completely compatible with 6510 (in fact, the 6502 is the parent of the family of microprocessors of which the 6510 is part of !). So at least in theory, any routine written for the 6510 can be executed on any BBC or other compatible machine.

But obviously, even if you don't use any of the Kernal ROM routines, there are still other difficulties to be considered. For example, the screen locations of the BBC computer are at a different place in memory than those of the C64.

Nevertheless, before using ROM routines you should ask youself, if the program needs to be portable.

Using ROM Routines

My advice is initially to kit yourself out with a decent disasembly of the Commodore 64 Kernal and the Basic ROMs.

The one which I use like a Bible is called *What's Really Inside The Commodore 64* and written by Milton Bathurst. It has the advantage of having lots of remarks (some disassemblies I've seen haven't got any remarks!) and it's published by DataCap, 12 Trixhai, B-4545 Feneur, Belgium. (I bought my copy from Boots.)

Studying such a disassembly thoroughly is a very good way of learning how a professional machine code program is written.

Secondly, it tells you all about the routines in your Commodore ROM. This allows you to use ROM routines not only from the starting points which are listed in many books and magazines (and also in this article), but you can also use ROM routines in your own way, maybe like nobody else has used them before.

Calling a ROM routine

Most ROM routines are called with a JSR instruction, because the majority of them are subroutines and end with an RTS instruction. This returns the program flow back to your own routine.

Before calling a ROM routine you have to know which registers will be used by it. Then, if you need any of the values later on, you have to make provisions to save them.

As we know there are two ways of saving a variable. Transferring it into the accumulator and then pushing the contents of the accumulator onto the machine stack or saving the variable in an address (if possible, a zero page address).

I prefer saving in memory, because it prevents the lethal bugs associated with the intricacies of the machine stack and I know at all times where everything is. Furthermore, I can recover things from that variable as often as I like without having to worry about pulling priorities.

It is important to be aware, which memory locations are used by the ROM routine you want to call. This

is especially important when using a zero page location. You then have to know the exact calling address of the ROM routine you want to use.

Although this may seem obvious, some ROM routine listing calls give the address of the vector and not the proper start of the routine.

For example, one of the most frequently used ROM routines is called CHROUT, which sends an ASCII character contained in the accumulator to the current device (mostly the screen). This can be called from four different points: $FFD2, $F1CA, $AB47 and $E716.

The start of the routine proper is $E716 and is the best one to use if you don't want to waste any time. $AB47 does some error checks before jumping to the routine proper. $F1CA is where the routine starts when it is called from the vector at $0326.

Finally, $FFD2 is the vector on top of the computer which in turn uses the vector at $0326, which again starts the routine proper at $F1CA.

ROM Typewriter

To give you a practical demonstration of how ROM routines can be used I have redesigned our little wordprocessor program to work entirely with ROM routines. This gives you a demonstration of some of the most often used ROM routines of the C64.

Also I have added an extra facility which gives a printout of the current line number and column number at the top of the screen. This makes the program more like a wordprocessor.

Figure 1 shows each ROM routine I have used, in detail and what you have to do to properly call them.

Figure 1 – ROM Routines

Print ($E716)
Prints any ASCII character onto the screen. Also executes non-printable characters such as delete, cursor movements, cursor home etc. All registers are saved at the start of the routine and recover again at the end, including the content of the accumulator. So you don't have to bother about saving anything.

Printstr ($AB1E)
This routines prints a whole string of characters (maximum = 256 characters!), including non-printable characters at the current print position. Before entering put the low byte of the start of the string in memory into the accumulator and the high byte into the Y-register. Zero is used as the end-marker of the string. Don't forget it! All registers are corrupted by the routine and have, if necessary, to be saved beforehand.

Printno ($BDCD)
Prints a 16-Bit NUMBER at the current print position whose low byte is contained in the accumulator, while the high byte should be in the X-register. This is used by basic to print line numbers on the screen. All registers are corrupted by this routine.

Plot ($FFF0)
Plots the current print position. The current print position is contained in the system variables $D3 (column) and $D6 (line). If the carry flag is set, transferred from those variables into the X- and the Y-register. If the carry flag is clear, the values in the X-register (column) and the Y-register (line) are plotted onto the screen, that is, made into the current print position. All registers are corrupted by this routine.

Getin ($FFE4)
Gets the value of the last key out of the keyboard buffer and loads it into the accumulator. If no key has been pressed, 0 is loaded into the accumulator. All registers are corrupted by this routine.

Cls ($E544)
Clears the whole Commodore screen and places the current print position to the top of the screen. All registers are corrupted by this routine.

Let's now look at the most important points of the program which you'll find listed as always at the back of the magazine.

In lines 340-350 I turn the system cursor on by loading the system variable $CC with 0. If it should contain a number larger than 0, the cursor would be turned off.

The main loop of the program consists of testing the keyboard, exiting from the routine if F1 has been pressed, printing the ASCII character on the screen or executing a non-printable character, such as delete, updating the line number if a new line has been started and, finally, updating the column number.

Since, as far as the operating system is concerned, each line consists of 80 characters, even though the screen can only portray 40 character lines, we have to make an adjustment to the next 40 column line before character 39 has been printed. This is done in lines 570-650.

The ASCII character contained in the accumulator is saved on the machine stack. Then the current row as contained in system variable $D6 is incremented and the beginning of the line plotted back to column zero. Finally, the ASCII character is recovered into the accumulator. Then the ASCII character is printed.

After this the current line number, contained in system variable $D6, is saved in 251 and the current column number, contained in system variable $D3, is saved in 252. This is because the print positions have to be replotted in order to print the line number, and column number at the top of the screen.

Next the current line number is printed at the top of the screen. Since the header takes up five lines, 5 is subtracted from the current row as contained in $D6 (lines 880-930). Next the current column number is printed at the top of the screen.

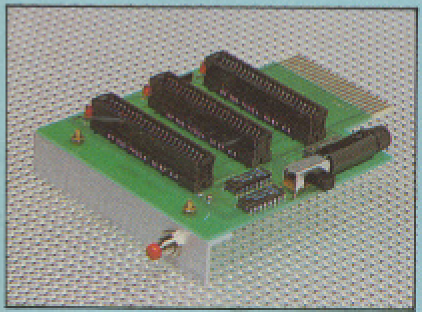
If a new line has been started, the former column number is blanked out with two spaces, otherwise one would get column 11, instead of column 1 (lines 980-1090).

Finally, the current print position is recovered from 251 and 252 and replotted. Then the routine loops back to get the next keypress.

The program as it stands has several drawbacks, such as the deletion routine not working properly from one line to the former. But I'll hope all you people out there will get busy mending these insuffiencies and make a fully working program out of it.

See listings on page 73

DATEL ELECTRONICS



3 SLOT MOTHERBOARD

Save wear and tear on your Expansion Port

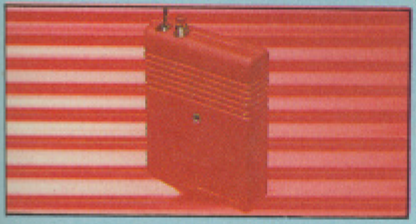
- ❑ Accepts 3 cartridges.
- ❑ Onboard safety fuse.
- ❑ Switch in/out any slot.
- ❑ High grade PCB.
- ❑ Fully buffered.
- ❑ Reset button.

ONLY £16.99

DATA RECORDER

- ❑ Quality Commodore compatible data recorder.
- ❑ Pause control.
- ❑ Suitable for 64/128.
- ❑ Counter.
- ❑ Send now for quick delivery.

ONLY £24.99



SMART CART™

Now you can have an 8K or 32K cartridge that you program like RAM then acts like ROM!

- ❑ 8K or 32K pseudo ROM.
- ❑ Battery backed to last up to 5 years (lithium battery).
- ❑ Simply load the program you require – then flick the switch. The cartridge can then be removed just like a ROM cartridge.
- ❑ Make your own cartridges including autostart types – without an EPROM burner.
- ❑ Can be switched on or off board via software.
- ❑ I/O 2 slot open for special programming techniques.
- ❑ 32K version has 4 x 8K pages.
- ❑ Some knowledge of M/C is helpful – but full instruction are provided.

8K VERSION £14.99
32K VERSION £29.99

SMARTCART UTILITIES

A series of utility programs for use with Smartcart. Simply load in the program and you have a powerful cartridge. When you have a different requirement – load in another program and you have a new cartridge. The process takes seconds, and can be repeated any number of times.

We intend to release a range of programs. The first available are:

DISKMATE II

❑ all the features of Diskmate II (see ad).

Loaded in seconds – with full instructions. **ONLY £9.99**



PRINTER IV

Now you can turn your MPS 801 into 4 printers in one!!

- ❑ Alternative character set ROM – fitted in seconds.
- ❑ All four sets have true descenders.
- ❑ 100% compatible with all software.
- ❑ Descender.
- ❑ Eclipse.
- ❑ Scribe.
- ❑ Future.
- ❑ Choose any set at the flick of a switch.

"For anyone who uses their printer often this chip is a must. It's quick and easy to use and the end results are impressive" – Your Commodore, Jan 87.

ONLY £19.99

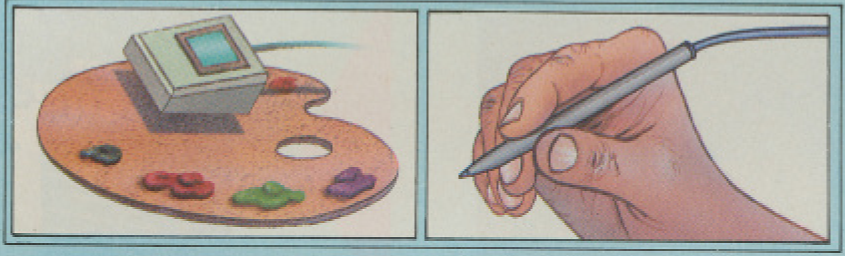


TURBO ROM II

Turbo Rom II is a replacement for the actual kernal inside your 64. It provides superfast load/save routines.

- ❑ Loads most programs at 5-6 times normal speed.
- ❑ Saves at 5-6 times normal.
- ❑ Improved DOS support including 10 sec format.
- ❑ Programmed function keys:- load, directory, old, etc.
- ❑ Return to normal kernal at flick of a switch.
- ❑ FCOPY – 250 block file copier.
- ❑ FLOAD – special I/O loader.
- ❑ Plus lots more.
- ❑ Fitted in minutes – no soldering usually required. (On some 64's the old ROM may have to be desoldered).

ONLY £14.99



BLAZING PADDLES

A complete lightpen/graphics illustrator package.

- ❑ A fully icon/menu driven graphics package of a calibre which should cost much more.
- ❑ Complete with a fibre optical lightpen system for pin point accuracy.
- ❑ Multi feature software including:
  - ● Range of brushes ● Airbrush ● Rectangle ● Circle ● Rubberbanding ● Lines ● Freehand ● Zoom mode ● Printer dump ● Load/save ● Advanced colour mixing – over 200 hues!!
  - ● Cut and paste allows shapes/windows/pictures to be saved to/from tape/disk.
- ❑ Blazing Paddles will also work with many other input devices including; Joysticks, Mice, Graphics Tablets, Trackball etc.
- ❑ Pictures can be saved from Action Replay and edited with Blazing Paddles.

Complete Blazing Paddles & Lightpen **ONLY £24.99**

Blazing Paddles available separately for **£12.99**

TOOLKIT IV

The ultimate disk toolkit for the 1540/1541

- ❑ A disk toolkit is an absolute must for the serious disk hacker. Toolkit IV has more features than most for less.
- ❑ DISK DOCTOR V2 – Read and write any track and sector including extra and renumbered tracks. Repair damaged sectors. Look underneath read errors.
- ❑ HEADER/GAP EDITOR – Decodes and displays ALL header information including off bytes and header gap. Rewrite the entire header and header gap. Renumber sectors. Also edit any sector tail gap.
- ❑ DISK LOOK – Sort directory. Recover lost files. Display file start/end addresses. Disassemble any file program directly from the disk to SCREEN or PRINTER including undocumented opcodes. Edit Bam. Much, much more.
- ❑ FILE COMPACTOR – Can compact machine programs by up to 50%. Saves disk space. Compacted programs run as normal.
- ❑ FAST DISK COPY – Copy an entire disk in 2 minutes or less using single 1541.
- ❑ FAST FILE COPY – Selective file copy. Works at up to 6 times normal speed.
- ❑ FORMATTER – 10 second format an entire disk or format any individual track or half track 0 to 41. Redefine any of 30 parameters to create or recreate unique disk formats.
- ❑ ERROR EDIT – Quickly find and recreate all read errors including extra and renumbered tracks or sectors and half tracks from 0 to 41. Even recreates data under errors and allows you to redefine any necessary parameters.

ONLY £9.99

RAM DISK

- ❑ Turn your Smart Cart into a 32K RAM/disk.
- ❑ 32K of instant storage area for files/programs.
- ❑ Load/save instantly.
- ❑ Disk type commands: load, save, directory, scratch.
- ❑ Program data retained when computer is switched off!
- ❑ Full command set with instructions.

ONLY £9.99

ROBOTEK 64

Model & Robot Control made easy

- ❑ Robotek 64 is a comprehensive hardware/software package which enables you to connect your 64/128 to the outside world.
- ❑ 4 output channels – each with onboard relay
- ❑ 4 input channels – each fully buffered TTL level sensing.
- ❑ Analogue input with full 8 bit conversion.
- ❑ Voice input for voice control.
- ❑ Software features:- test mode/analogue measurement/voice activate/digital readout etc.
- ❑ Excellent value! **ONLY £39.99**

including hardware/software/mic etc. (State tape or disk)



DIGITAL SOUND SAMPLER

- ❑ The new sampler allows you to record any sound digitally into memory and then replay it with astounding effects.
- ❑ Playback forwards/backwards with echo/reverb/ring modulation etc.
- ❑ Now with full sound editing module to produce outstanding effects.
- ❑ Full 8 bit D to A and ADC conversion.
- ❑ MIDI compatible with suitable interface. (I.e. Datel unit for £29.99, see ad).
- ❑ Live effects menu includes real time display of waveforms.
- ❑ Line in/mic in/line out/feedback controls.
- ❑ Powerful sequencer with editing features.
- ❑ Load/save sample.
- ❑ Up to 8 samples in memory at one time.
- ❑ Tape or disk (please state).
- ❑ Complete software/hardware package **£49.99**
- ❑ Com-Drum software is available separately at **£9.99** to turn your Sampler II into a Com-Drum system as well as a sampling system.



COM-DRUM

Digital Drum System

- ❑ Now you can turn your computer into a digital drum system.
- ❑ Hardware/software package.
- ❑ 8 digital drum sounds in memory at one time.
- ❑ Complete with 3 drum kits.
- ❑ Real drum sounds – not synthesised.
- ❑ Create superb drum rhythms with real and step time.
- ❑ Full editing.
- ❑ Menu driven.
- ❑ Output to hi-fi or through tv speaker.
- ❑ Load/save facilities.

ONLY £29.99 (state tape or disk)

COM-DRUM EDITOR

- ❑ 24 drum sounds supplied on disk to enable you to construct your own drum kit.
- ❑ Re-arrange sounds supplied with a Com-Drum to make new kits.
- ❑ With sound sampler hardware you can record your own kits.
- ❑ Load/save facilities.

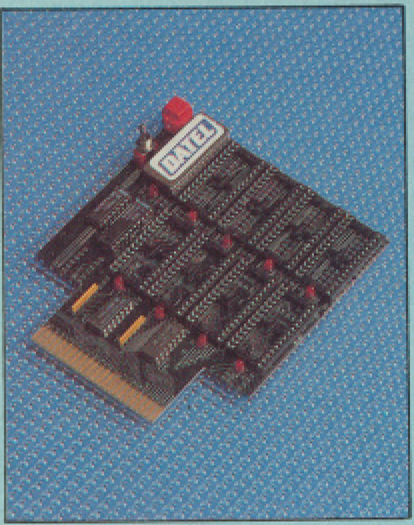
ONLY £4.99 disk only

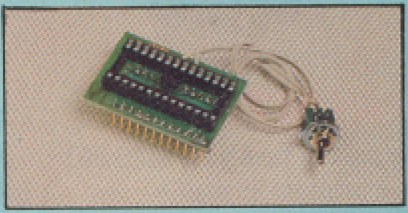
DATEL ELECTRONICS

256K SUPEROM EXPANDER

- Now you can select from any of 8 32K EPROMs instantly.
- 8 sockets to accept upto a 32K EPROM in each.
- On board operating system – no programs to load.
- Program your own EPROMs using our EPROM programmer.
- No need to have loads of cartridges – just make a selection from the Superom menu.
- Directory of utilities on power up.
- Fully menu driven on power up.
- Select any slot under software controls.
- Unique EPROM generator feature will take your own programs – basic or m/c and turn them into autostart EPROMs. (EPROM burner required).
- Accepts 2764/27128/27256 EPROMs.
- On board unstoppable reset.

ONLY £29.99



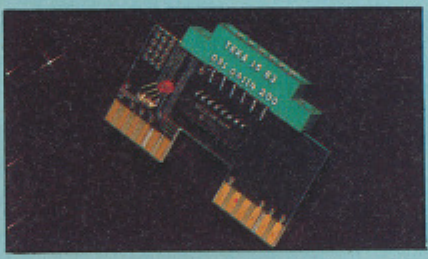


4 WAY KERNAL BOARD

- This board fits in place of the kernal in your 64 and accepts a 16K or 32K replacement kernal giving 2 or 4 different operating systems.
- Just flick the switch supplied to select between systems.
- This is a carrier only – ready to accept your own chip.
- Now you can have all your different systems available at one time.

ONLY £8.99

5 WAY BOARD **ONLY £12.99**



CLONEMASTER™

- Makes tape to tape back-ups.
- Works with even Turbo Loaders etc.
- Requires access to two CBM type data recorders.
- Makes perfect copies.
- Very simple to use.
- LED level indicator.

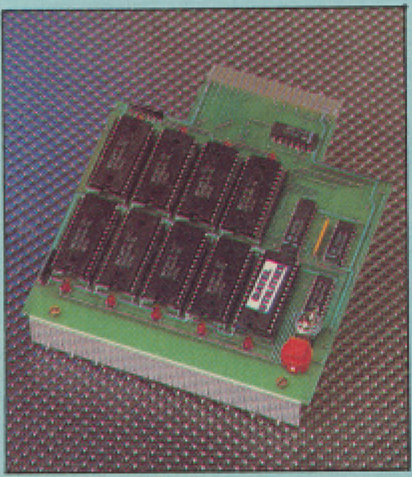
ONLY £9.99 POST FREE



RESET CARTRIDGE

- Unstoppable reset button.
- Resets even so called "unstoppable" programs.
- Add pokes from magazines etc.
- Fits in cartridge port.
- Simply plug in.

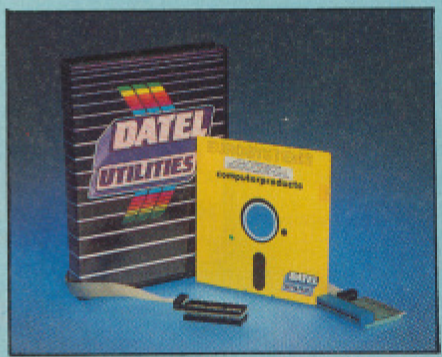
ONLY £5.99



DUPLIKATOR™

- Copies whole disk in one pass – only one drive required!
- Makes back-up copies in eight seconds!!
- Duplikator is the fastest, most efficient and easiest to use disk copying system ever conceived.
- Comes complete with 256K on-board ram and it's own operating system – no software to load.
- Makes multiple copies from one original.
- Full disk verification during back-up process.
- Copies standard software as well as errors 21-29 and upto 40 tracks.
- Full disk error check in eight seconds.
- Full disk verify against ram in fifteen seconds.
- A must for clubs, user groups etc. How else can you copy over 250 disks and hour for less than £100.
- Comes complete with on/off switch and reset button.
- Fitted in minutes – no soldering usually required.

ONLY £89.99



DEEP SCAN BURST NIBBLER™

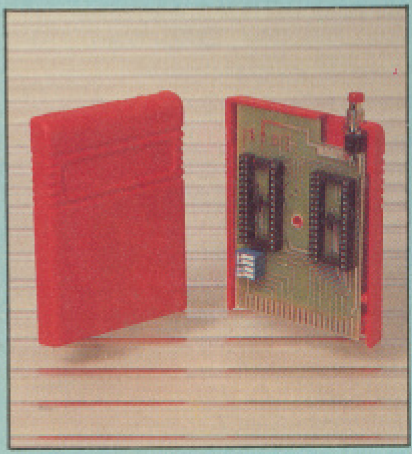
- The most powerful disk nibbler available anywhere, at any price!
- Burst Nibbler is actually a two part system – a software package **and** a parallel cable to connect the 1541/1570/1571 to 64/128 (state type).
- What gives Burst Nibbler its power? Conventional nibblers have to decode the data from the disk before it can transfer it using the serial bus – when non standard data is encountered they are beat. Burst Nibbler transfers data as raw GCR code via the parallel cable without the need to decode it so you get a perfect copy of the original.

● Will nibble upto 41 tracks. ● Copy a whole disk in under 2 minutes. ● Full instructions. ● Regular updates – we always ship the latest. ● Fitted in minutes – no soldering usually required. ● Full or ½ tracks. ● No need to buy parallel cable if you have Disk Demon/Dolphin etc. ● Cable has throughbus extension for other add ons.

- Whether to choose FastHack'em or Burst Nibbler? Fast Hack'Em is unbeatable value as an "all-rounder" – with nibblers, 1 or 2 drive copy, format, file copy, 1571 copy etc. etc., so if you have a more general requirement perhaps FastHack'Em is for you. Burst Nibbler is a pure nibbler second to none, for the reasons stated. So if it's just making backups you are interested in, there is no other prodyct to beat it.

ONLY £24.99 COMPLETE

SOFTWARE ONLY £12.99 CABLE ONLY £14.99



16K EPROM BOARD

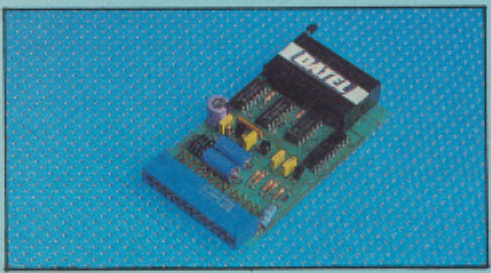
- Accepts 2 x 8K EPROMs.
- Switchable to configure as 2 x 8K or 16K or off.
- On board unstoppable reset.
- Full instructions.

ONLY £8.99

DISK NOTCHER

- Quickly and easily double your disk capacity.
- Use both sides of your disks.
- Simple to use.
- Takes seconds.

ONLY £4.99



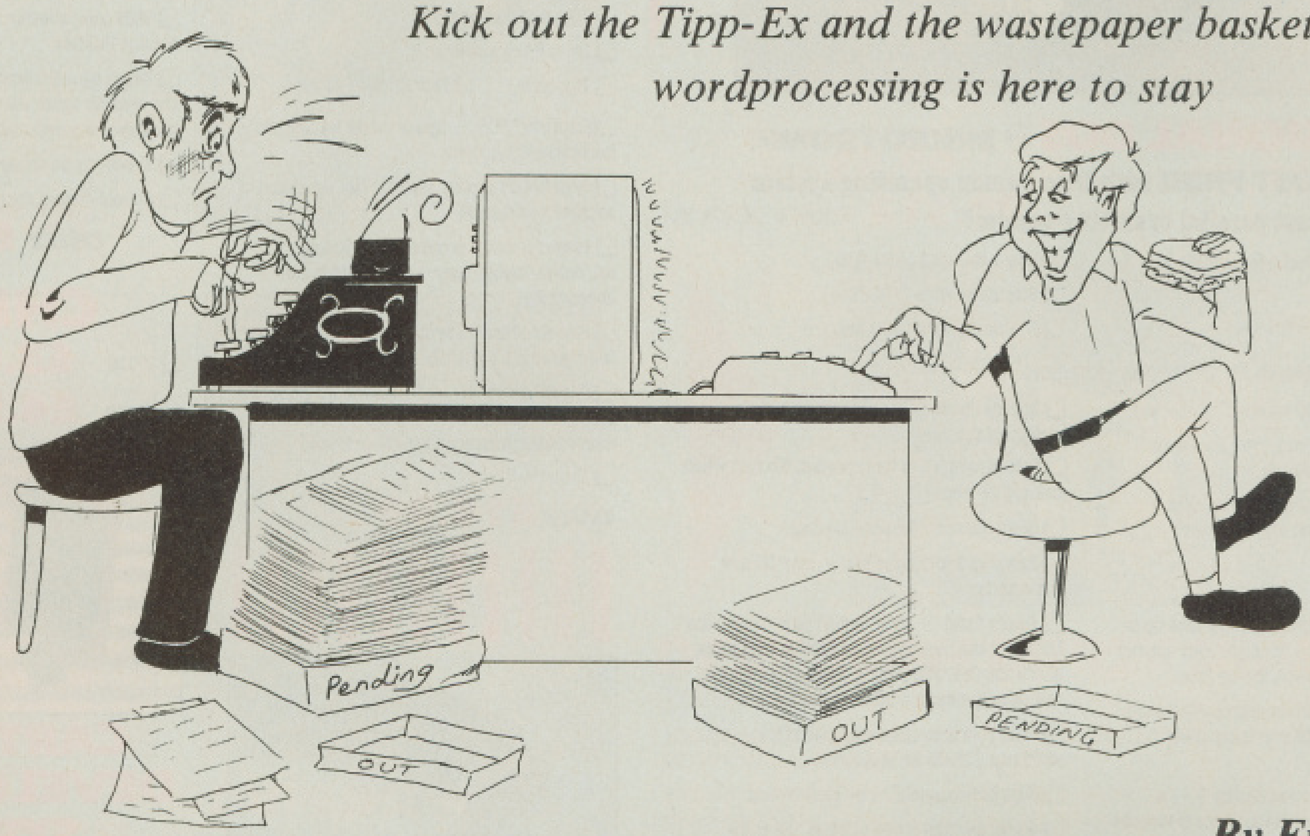
EPROMMER 64™

- A top quality, easy-to-use EPROM programmer for the 64/128.
- Fully menu driven software/hardware-package makes programming/reading/verifying/copying EPROMs simplicity itself.
- Will program 2716, 2732, 2764, 27128 and 27256 chips. 12.5, 21 or 25 volts.
- Fits into user port for maximum compatibility with cartridges/Superom Board etc.
- Full feature system – all functions covered including device check/verify.
- We believe Eprommer 64 is the most comprehensive, most friendly and best value for money programmer available for the 64/128.
- Ideal companion for Superom Board, Cartridge Development System, our kernal expanders or indeed any EPROM base project.
- Comes complete with instructions – plus the cartridge handbook.

ONLY £39.99 COMPLETE

The Moving Cursor Writes

Kick out the Tipp-Ex and the wastepaper basket, wordprocessing is here to stay



By Eric Doyle

In the Tenth Century, wordprocessing was a thing of scorn. In a review for the Rubaiyat, Omar Khayyam wrote, 'The moving finger writes; and, having writ, moves on: nor all thy piety nor wit shall lure it back to cancel half a line, nor all thy tears wash out a word of it.' Not very user friendly. When Edward Fitzgerald translated the Rubaiyat into florid verse, the age of the typewriter was dawning and the writing was on the wall for the moving finger system.

Today the wordprocessor has supplanted the typewriter and few authors pummel plattens to beat their unruly prose into a semblance of literary worth. The wordpro revolution means that corrections, updates and re-arrangements can be made with ease and there are few people who would not benefit from its use.

Even for a humble job application, the power of the wordpro can be harnessed to produce faultless text and the days are numbered for liquid paper correction fluid. Even though the use of the wordpro is so well accepted, there is still the need to show what is essential, what is desirable and which features are rarely needed.

What is a Wordpro?

A wordprocessor is basically like a typewriter. It allows text to be typed in at the keyboard of a computer to give a hardcopy on paper. Where a wordprocessor differs lies in the fact that it acts as an interface between the keyboard and paper, expanding the possibilities far beyond a typewriter's wildest dreams.

The essential features of a wordprocessor can be broken down into three categories. There is the actual entry of text, the ability to specify layout, and the facility to access a printer and storage device. In addition to these essentials, the use of a spelling checker can make life easier and, for a very few people, integral modem software can send the completed text to any printer in the world.

Editing Facilities

The basic requirements of text editing is to be able to type in characters at a comfortable speed, see the text displayed on the screen, and to correct errors before the text is printed. All of this can be achieved with many of the microchip typewriters currently available but a wordpro can do more.

Nobody's perfect so the wordpro should allow the user to scan through the text making deletions and insertions. Major rewriting and the addition of large blocks of text ought to be possible.

Sometimes this modification process means that the whole structure of the document needs to be changed. In the bad old days this would mean that yet another ball of paper would whistle towards the wastepaper bin. With a wordprocessor it is usually possible to mark a block of text and move it from one place to another or to repeat it if it is a clause that is used several times. This saves a lot of retyping and consequently saves time and tribulations.

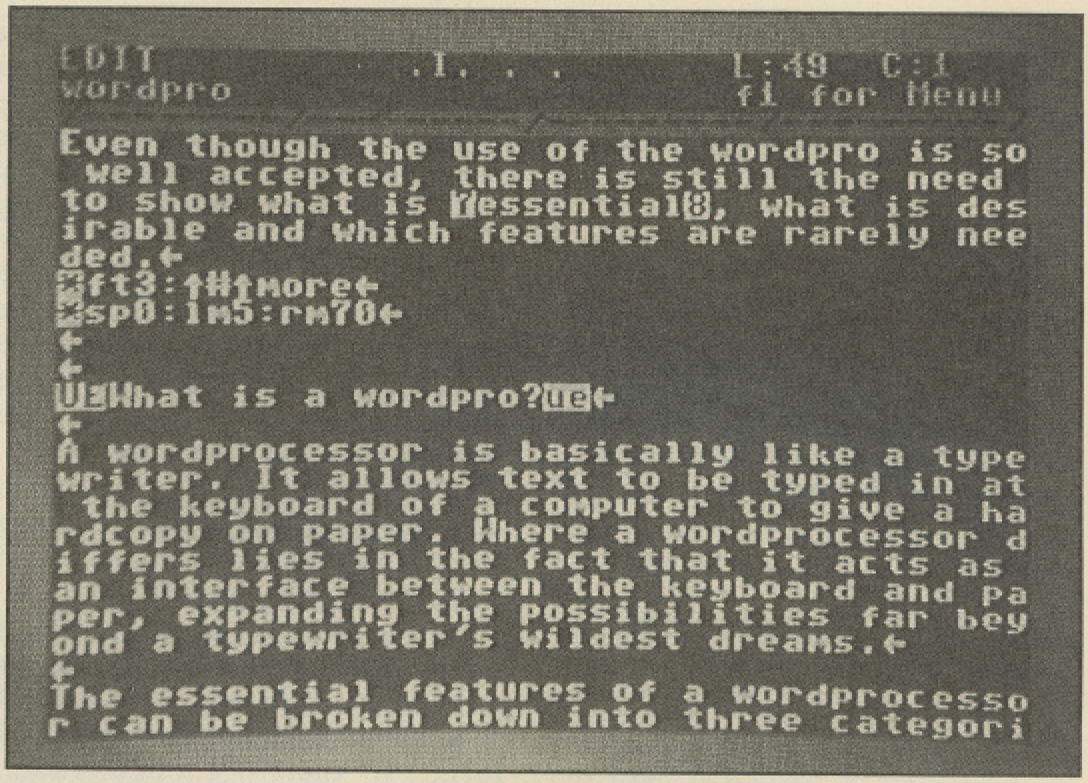
With large documents, the user may want to find a particular word or phrase and a search facility would be useful. Most wordprocessors contain a command line which in this case is used to allow the searched for text to be entered. The program will then search through the document for every occurence of the character string and highlight it. At each find the user will be required to indicate whether this occurence is the one desired. If it is not, the program will search for the next appearance of the string.

The query is needed because the program is not intelligent in the way we are. If the desired string was the word 'where', the computer would highlight the string in words like 'wherefore', 'wherever', 'everywhere', and so on as well as the word itself.

Added power can be added to this command if a replace function is added. Now each time a word is found it can be changed to another specified word. As an example, the word 'Basic' referring to the computer language may be required in capitals. To search for each occurrence would be tedious but the search and replace function will do most of the work for you. Once again the user must okay any changes because the word basic may have been used in its true sense, of meaning fundamental in which case a change is not necessary.

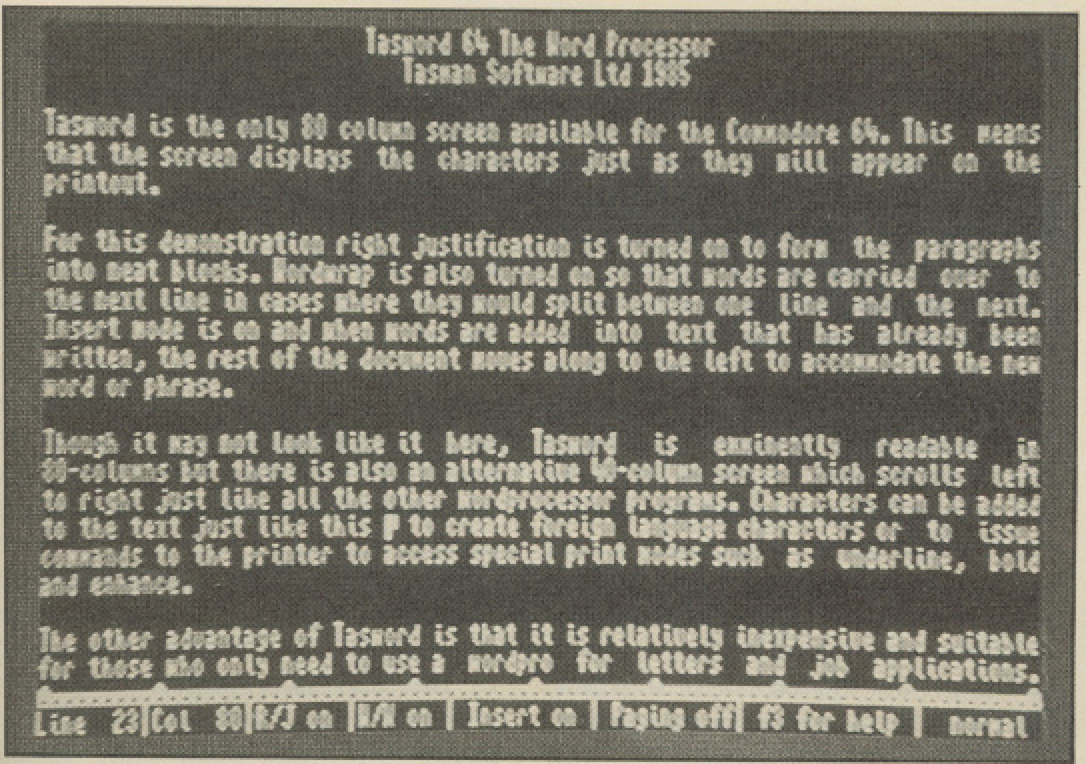
Formatting

On a printer the normal paper size is A4 and a line width of between 60 and 80 characters is normal. The computer screen of the C64 and Plus4/C16 is only 40 characters wide so it is difficult to show the shape of the finished text as it will appear on the printed page. To combat this many ingenious methods have been tried.



Tasman's Tasword 64 program uses a special small character set which converts the screen to an 80 character display. Even with the subsequent halving of horizontal resolution, the characters are still readable but the program also allows a 40 column screen display to be used to clarify any areas which may be a little difficult to read.

Most systems use the sideways scroll which allows the viewer to scroll across the page column by column. This is as though the screen was a window past which the page can be scrolled left and right or up and down. At any one time, the view is part of the document measuring 40 characters wide and 22 or more lines deep.



Homeword was a cumbersome program to use but had an excellent feature which showed the formatted page in the form of a window which represented each character by a pixel dot. The general shape of the document could then be seen and altered to give a pleasing aspect.

Users of the C128 with a 1901 monitor can access the 80 column screen and many wordpros have special versions included with the normal C64 version to enable this facility to be used.

After treatment of text is the main area where one wordprocessor can score heavily against another. The minimum requirement is to be able to specify the length and width of each page but there is a plethora of refinements which can best be separated into two categories of style and form.

In both cases the commands which achieve the desired effect are produced by special character strings, known as embedded commands because they form part of the text entry:

*it; this is an example of an embedded command.

When the formatted viewing screen is called up, these commands are

executed rather than being displayed, so the raw text is revealed:

This is an example of an embedded command.

The theory only proves true to a limited degree because WYSIWYG is not available. WYSIWYG simply means What You See Is What You Get and refers to a screen display which looks exactly the same as the printed-out page. For many reasons relating to efficient use of memory and character definition, it is not possible to utilise WYSIWYG displays on the Commodore. Instead a program may highlight the characters which will receive special attention by reversing out the display or by some other device.

Stylish Characters

Modern printers allow various styles of characters to be used within a single document. Enlarged characters, italics, NLQ, different typefaces should all be readily accessible to the user.

The overall appearance of the document can be affected by the effect known as justification. This means that characters can be aligned to the left column leaving the right edge looking ragged as each line has a different number of characters. Right justification can also be employed which means that extra spaces are added to successive lines to make all the lines the same width so that a uniform block of text appears. *Your Commodore's* pages right and left justified. Glance through a few paragraphs of this text and you can see the effect that this has on individual lines.

If the line width is short, the justification process can result in two words being separated on one line by several spaces if the third word is too long to fit beside them. The way that the third word is pushed onto the next line is known as wordwrap and avoids words being split in peculiar ways. Let's see what happens when I give the typesetter a headache by using a very long word.

A long word like antidisestablishmentarianism causes problems on a line which has a maximum character count of about 38 characters per line. There are two ways in which this can be handled by the typesetter. First, after the four short words a new line is started for the long word. This means that sixteen characters are ranged across a line which normally contains twice as many characters. The result is a gappy line. The second method of treating this situation is to break the longer word but there is a convention to be observed. Words should only be broken down into syllables. The computer program cannot tell where syllables occur so some programs have a device known as a 'soft hyphen'.

Soft hyphens are characters which may or may not be displayed according to a single rule. Take the word 'somewhere' as an example.

If a soft hyphen is used the word is written as some-where. When the word some-where appears at the end of a line it splits into its hyphenated form. If there's enough room, somewhere isn't split and the soft hyphen is ignored by the program.

The choice of whether the soft hyphen facility is useful or not depends on the user. To create every word in its hyphenated form would be a bind but to be able to use it when previewing the page before printout can avoid problems occuring if changes are made elsewhere in a paragraph which pulls the word back into the middle of the line. A hard hyphen would remain but the soft one disappears.

When writing a letter, my text looks just as is does in this article; but when submitting this article my original text was double spaced. This means that instead of issuing a single carriage return at the end of the line, two returns were sent leaving a blank line between each printed line. This is done so that there's plenty of space for marking up special features such as *italics* or for correcting grammar and misspelt words.

Any wordprocessor for my use must have the facility to double space lines and the one I use can actually triple space as well.

On the monitor each character is given equal spacing. Some characters like the letter 'm' fill the space comfortably but letters like 'i' and 'l' leave large gaps between themselves and neighbouring characters. Examine the text you're reading and you should observe that the spaces between letters are approximately the same on any given line. This means that characters next to the letter 'i' have been automatically moved closer together. This is known as proportional spacing.

The better printers can proportionally space letters so the best wordprocessors are given commands which allow access to this facility.

Patterns on a Page

The main features necessary to set up a page for print are the margins. These not only appear at the sides but also at the top and bottom. To add to this the page may not be A4 in height. So there are five parameters which must be expressed: page length, left and right margins, and the bottom margin.

Left and right margins determine the number of characters per line. The normal value is 80 but there may be occasions when a wide platten printer is used and 132 characters or more can be fitted onto a line. On other occasions A5 paper may be required so a much smaller page width might be required.

The ability to specify long and short line widths is not the only feature required because the preview option needs to be able to range across this number of characters. If it only has a maximum of 80 characters, the preview screen is useless for wider documents.

Once a line width is set, it is far more interesting if the document has indented sections to highlight particular features. Some wordpros allow temporary margins to be set. This could be done by simply inserting spaces at the beginning of each line but these spaces are saved when the program is stored and means less space for the document.

There are also occasions when each paragraph may have a heading, correctly called a sub-heading. One way of doing this is to indent the following text to highlight each sub-head. To insert spaces on virtually every line would be boring especially when the power of the computer can be called upon to do this dull task for you.

Longer documents need page numbers and for some uses each page must have an identifying piece of text across the top or bottom of the page. It's a bind if you have to add this text each time a new page is started or finished. Even if the user accepts this limitation, the real problem becomes clear when a block of text is added to the text later which completely messes up the start and end of the pages following. Each page would

have to be laboriously altered. For this reason a header and footer facility should be available.

The header and footer is a command which allows a string of characters to be designated at the beginning of the text. This string will then be repeated at the top or bottom of each page and the facility normally allows a number character to be entered anywhere along the line. As each new page begins the page number is automatically incremented and inserted.

The final essential is a centring facility. To find the correct position for the start of a centred heading means counting the number of characters in the heading, dividing it by two and then subtracting this value from the maximum number of characters per line. A centring facility does this automatically but take care when using expanded characters or proportional spacing because I haven't found a program which copes adequately with these situations.

Hard Lines

Some documents are longer than the available memory allows. To create longer texts a linker can be useful. This permits a line to be added to a file which automatically searches for, loads and prints the next document in the chain.

When saving a document, there is a need for a save and replace function. The great disadvantage of a wordpro is that it stores the text as volatile electrical impulses which can be wiped out easily if the power supply is interrupted. Regular saving of the text is recommended if disaster is to be avoided. If the text is saved under a new name each time, the disk soon fills up with redundant files.

The alternative is to use the save and replace function to store an ammended document under the same name as before. The disadvantage with this is that the program often uses the inbuilt save and replace function of the disk drive. This is a bugged feature and can result in the loss of a file or the corruption of another file.

It's difficult to know if a wordpro uses save and replace or if it scratches the file before doing a normal save. It is better to use two filenames and address each on alternative saves.

The computer has a limited memory which is further reduced by the inclusion of the program itself. This means that computer storage space is at a premium and often a document will overrun the allotted memory space. Longer documents have to be stored over several files and a linker facility to chain them together makes life a lot easier. A special code inserted at the end of each document file combines the files into a long chain so that when each one finishes printing, the next part loads and prints automatically.

Special facilities to call up the directory and to send disk commands are useful especially when formatting a new disk for file storage. Equally, the ability to send special codes to the printer lengthens the life of a wordpro because printer technology is advancing rapidly with new commands and features being added with each new machine.

Printers fall into various categories. Commodore has its own codes, Epson have a different set, compatibles may vary from the Epson standard in small but important ways, and there are still many manufacturers doing their own thing where codes are concerned.

There are three ways of combatting the diversity needed. The cowards way out is to support one machine type and let owners of the others battle with the problem by sending out character string commands themselves or relying on an external interface to cope with the problem. The workman system is to incorporate an interrogator which asks for the information for creating special effects. This information is then stored as a block in memory which asks for the information for creating special effects. This information is then stored as a block in memory which can be saved and loaded when required. The heroic, idiot-proof system is my favourite where a front-end is added to the program asking what type of printer is being used. According to the response, a ready made file is loaded from the master disk and the system is up and running from day one. This system also incorporates the workman-like facility for rare, exotic species of printer.

The more expensive wordpros usually allow user defined codes to be issued. Special keys are reserved and these can be pressed either to send extra codes to printers or to issue strings of wordprocessor commands. In all cases, colours of screen, border and characters can be customised and printouts should be easily aborted for those times when some error becomes obvious at a late stage.

Extra Powers

Many wordpros are now accompanied by a spelling checker which will sift through the text and query any word which it doesn't recognise. The best checker that I've used is the one accompanying Logotron's Writer 1295. It is both fast and has a good initial vocabulary which can be edited and expanded. Some spelling checkers are laboriously slow and inflexible incorporating words which, though difficult to spell, are hardly ever used in common parlance.

A spell checker is a utility and not a magic wand. For example, I seem to have developed a nasty habit of writing 'their' when I mean 'there'. I know the difference but just can't seem to break the habit. No spelling checker will correct grammatical or syntaxial errors so a command of the English language is still a requirement of the user.

For journalists like myself, a word or character count is essential. Many of the programs go beyond this and give paragraph and sentence counts.

One feature which can be useful if your work requires the manipulation of figures is a built-in calculator. Normally these are limited in accuracy to two decimal places for financial calculations.

The final facility is one which is purely a business or club feature. When a circular is to be sent as some sort of mail shot, it can be extremely time consuming changing addresses and names to personalise these communiques. A mail merge function allows keywords from a specially constructed file to be substituted where indicated in the text allowing a high degree of customising to take place.

Most people will at some time be subject to the cunningly phrased Reader's Digest special offer using cosy phrases like: 'the Doyle family has been specially selected', or 'imagine the expression on the faces of your friends, Eric'. Look carefully at the construction of the messages and you'll soon realise that you're a victim of a mail shot. How many children receive special offers for themselves and their wives or husbands! How many sub-18 year olds have received the chance to win a car of their own. Computer intelligence is limited!

The mail merge is almost a standard in small but important ways, but some also link through to databases which saves the unattractive prospect of creating two such files. The C128 version of Superscript can reside in memory alongside Superbase. For mail merge processing this has a distinct advantage.

What to Look For

When buying a wordpro the first consideration is to sit down and work out which facilities you need, anticipate those which may be of use in the future and then start ploughing through the range to find a system that suits your needs and your pocket.

Just because the program has everything that you need doesn't mean that it's definitely the one for you. If possible ask for a demonstration because although most wordpros offer similar basic features the way that the end effect is achieved may not be appealing.

Check that the program supports the printer that you intend to use. A bad selection here would be an expensive error.

Some wordpros allow files to be saved as ASCII files. The advantage of these programs is that commonly used files can be transferred to a new wordpro if a change in system is considered and files can be transmitted to any computer through a modem.

If modem linking is an essential feature, Paperclip 128 has built-in terminal software. A comprehensive program such as this means that program hopping between wordpro and terminal can be avoided.

Laziness should be avoided. It's easy to say that buying the wordpro that was everything, will cover all future developments but the manual will also be complex. This makes it difficult to sift out the essential features that you require.

For most wordpros the manual is an essential companion. Eventually the commonly used codes will become familiar but the lesser used commands will have to be looked up. A manual which has a potted guide to commands can be a boon. Superscript has a very useful menu display which can be called up to readily access all of the commonly used features. A program with a Help screen can also be a good buy.

SuperScript

Supplier: Precision Software, 6 Park Terrace, Worcester Park, Surrey KT4 7JZ.
Price: C64 disk £24.95
C128 disk £29.95
Comments: This is my favourite because it's the most user-friendly system. It has a wide range of printer files, a reasonable spelling checker and a calculator. The readily accessible command menu is brilliant. C128 version has 40/80 columns and can reside in memory with SuperBase.

Easy Script

Supplier: Commodore Business Machines, Commodore House, The Switchback, Gardner Road, Maidenhead, Berkshire SL6 7XA
Price: C64 disk £39.50 (was a freebie with the 1541).
Comments: This is really a less user friendly SuperScript without the menu or the spelling checker. The price makes SuperScript the better buy.

Writer 1295

Supplier: Logotron, c/o Vector Services, 13 Denington Road, Wellingborough, Northants NN8 2LR
Price: C64 disk £12.95
Comments: Don't be fooled by the price, this is an excellent new wordpro with the best spelling checker that I've seen. The features are worthy of a package costing twice as much.

Paperclip

Supplier: Was Ariolasoft; can still be found in shops.
Price: C64 and C128 disk £44.95
Comments: Undoubtedly an excellent package but inhibited by price. The C128 version has 80 column screen capability and modem terminal. Excellent 15,000 word Spell Checker is not available on all versions so buy with care.

Tasword

Supplier: Tasman
Price: C64 disk £14.95
Comments: The only true 80 column C64 wordpro but otherwise nothing startling. A good work horse.

Word Perfect

Supplier: Supersoft/ ASL Software, Winchester House, Canning Road, Wealdstone, Middlesex HA3 7SJ.
Price: C64 Disk £19.95, cassette £17.95
C16 disk £14.95, cassette £12.95
Comments: Word Perfect is a good basic wordpro and has the distinction of being the only one available for the C16. The printer interface is set up for Commodore only but special commands can be sent to other types of printer but no save facility is available to automatically resend these commands when the wordpro is used again.

VizaWrite/ Viza Classic

Supplier: Calco Software, Lakeside House, Kingston Hill, Surrey KT2 7QT
Price: Write C64 disk £39.95 cartridge £49.95
Classic C128 disk £59.95
Comments: The editor's favourite. Phenomenally expensive yuppie C128 package but has a full range of facilities. The Classic is only available for the 80-column mode. Help screens may be loaded from disk rather than referring to the manual.

Mini Office II

Supplier: Database, Europa House, 68 Chester Road, Hazel Grove, Stockport, SK7 5NY
Price: £16.95 (Ca) £19.95 (Disk)
Comments: This is a fully integrated system incorporating wordpro, database, spreadsheet, Business graphics, comms pack and label printer. Its sales have made it a blockbuster. The wordpro is surprisingly good for such a low cost system. If a full pack is what you need, you could buy worse at three times the price.

Cut'n'Paste

Supplier: Was Ariolasoft; can still be found in shops.
Price: C64 disk £21.95
Comments: Extremely user friendly but basic package. Main feature is the text transfer referred to in its title.

STRUCTURED PROGRAM DESIGN

Although the advantages of structured programming are often mentioned, there seems little information which explains how to achieve it! So for those of you who would like to improve the standard of your programs – read on!

By Derek Barrett

Traditionally, and I am the first to admit to having done this, home computer programmers often write their programs by sitting at the keyboard and just typing away. If a new idea for a routine comes into their head they slip in a GOTO and add the new piece at the end. This leads to 'Spaghetti' programs where GOTOs are sprinkled randomly throughout the code. Anyone who tries to read such a program often ends up in a mess trying to unscramble it.

This is not to say that the orderly use of GOTOs is wrong; sometimes the only way round a problem is to use them. It is only when they are used indiscriminately that they lead to confusion.

If you're used to designing your programs by flowcharts, you'll know that even in the early stages, you must think in terms of the smallest detail of code that will be needed in the final program. Constant redefinition and refinement is needed to the design before it can be run.

The technique described in this article allows you to start from simple 'root' ideas and refine each step as you progress through the design phase. The final design will look rather like an inverted tree, with the branches forming the paths of the design. That is why the technique is called *top down design.*

Each branch can be designed independantly and in any order. Often the easiest solution to a problem is to start by thinking of how to produce the output that will be required and then work backwards through the stages that are needed to produce that output.

Language Independant Code

An advantage of this method is that the designs you produce are completely portable and can be coded into any language on any computer. So whether you program in Basic, Pascal, Fortran or even assembler, the same rules apply. You can also use the technique for everyday decisions, totally unrelated to computer programming.

Think about this – you have just produced a brilliant program. A friend sees it and would like to run it on his machine. However his computer is not compatible, with a different dialect of Basic, and anyway he prefers to program in Pascal as he uses that at work. With top down programming there is no problem. Just give him a copy of the design and he can code it himself. This is much better than trying to redesign the program from your scrambled Basic.

The beauty of top down design is that no complicated new terminology or diagrams are used. If you can draw a simple rectangle you can start to design well-structured programs.

Right, let's get on with it. There are only three elements to structured design:

- Sequences
- Selection
- Repetition

Sequences

A sequence is the basic building block of top down design; each box indicates an action that will be carried out. The top box indicates the program title and is sub-divided into lower boxes which indicate the processes needed for the program to perform its function.

If you look at Figure 1 you'll see A is a sequence of B to F which are the main stages needed in a program to compare the latest reading of a car mileage with a previous reading and printing the difference.

At this stage of the design, the program has been broken down into manageable chunks with no attempt to define the detail that will be needed to code the problem. Imagine you are writing a book. Box A is the book title and B to F are the chapter headings. The contents of these chapters will be written later. Similarly boxes B to F will be broken down as the design continues until the final solution is reached. Each box contains one action only and the temptation to group ideas into one box must be avoided as it will lead to confusion.

Although Figure 1 is shown as being self-contained it could easily be part of a much larger program, for perhaps, a vehicle fleet servicing record. Very often a large program is best tackled by splitting it into sections that can be considered seperately.

For those of you more familiar to reading flowcharts the position of the linking lines from A to B, A to C etc. may seem strange, but you will see the reason for this when we expand the ideas further.

Selection

This is the second of the structures to be learnt and has the same function as IF...THEN in Basic. The only addition to our simple box is the addition of an 'o' in the upper right of the box.

Figure 2 shows a simple example of IF...THEN. You'll see that the actions are a sub-division of box A which serves as a 'Dummy' box i.e., it should be used as a comment line in the final code to aid legibility.

Box B is the IF part if the statement and Box C contains a line to show that nothing is to be done if the conditions of the IF statement is not met.

Box D forms the THEN part of the statement. You will notice that the instructions are written in pseudo code to make it easier to use the features of the language you will eventually code in.

Figure 3 expands this example to illustrate the IF...THEN...ELSE construction found in most languages. In this case there is an alternate action to be taken, with the ELSE part of the statement in box E. With this type of construction, flow is only allowed to go from A to B or A to C.

In other words, if the condition of B is TRUE, program flow will continue at D (C and E will not be carried out). Conversely if B is FALSE C and E will be carried out. You can see from this that C in Figure 2 really means ELSE DO NOTHING. A is a selection of EITHER B or C but not both. Figure 4 extends this to form the CASE statement. In this case flow will go to one only of B,C,D or E.

Multiple conditions can also be shown in the design (Figure 5) but you must be careful as it is easy to design a condition that can never be met, in which case the program could 'hang up' at that point. Box C can be coded using the ELSE IF construction if your language permits it. It is probably better, at least until you are confident in using the technique, to use the nested condition statements (Figure 6.)

Here boxes F,I and L must be included to allow flow to continue if the conditions are not met. Incidentally I have used letters adjacent to the boxes purely for ease when explaining the principle. Some software designers use nested numbers instead and I tend not to bother at all. It is entirely up to you.

Walking Through The Design

Before we go any further it is essential that you check the design thoroughly before you start to code it. The usual method is to 'walk' through the design, using dummy data, to ensure that it has the desired result and that no hidden traps have been put in.

The rule for reading a top down design is to start at the top and work down each branch, working from the left to the right.

Draw up a table of dummy data that will allow all the conditions of the stage to be tested. Now 'walk' through the design using this data to see where you end up.

We have entered the design at A, which is a sequence of B,C and D. Our tests are only concerned with the detail for C at this stage.

1st test. Task no.=3, Batch Total=200, Total Quantity < MAX. (MAX is defined elsewhere in the program.)

E is TRUE so go on to G
G is also TRUE so carry out Action 1 at J.

We have now reached the end of a 'branch' so we must go back to E. As this is an IF...THEN condition we cannot proceed to H or I so we exit back to C which is also an IF...THEN so we go on to D. This is vital to the concept of Top Down and believe me it is easier to do than describe.

2nd walk through

Task no.=3, Batch Total=350, Total

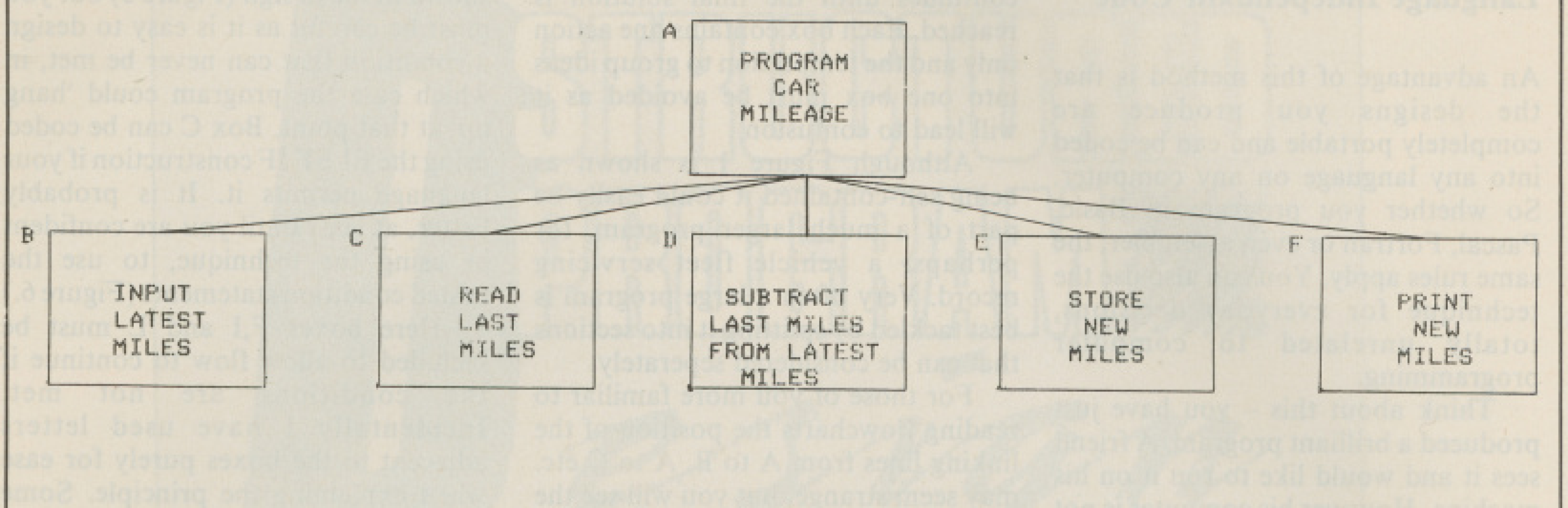


Figure 1 – example of sequence construction

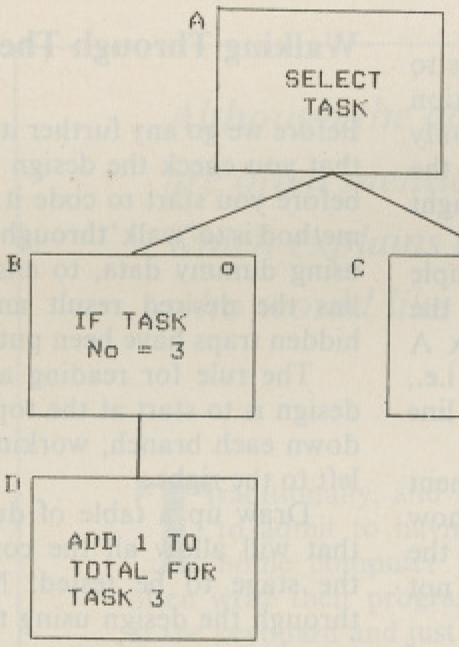


Figure 2 – selection with single choice

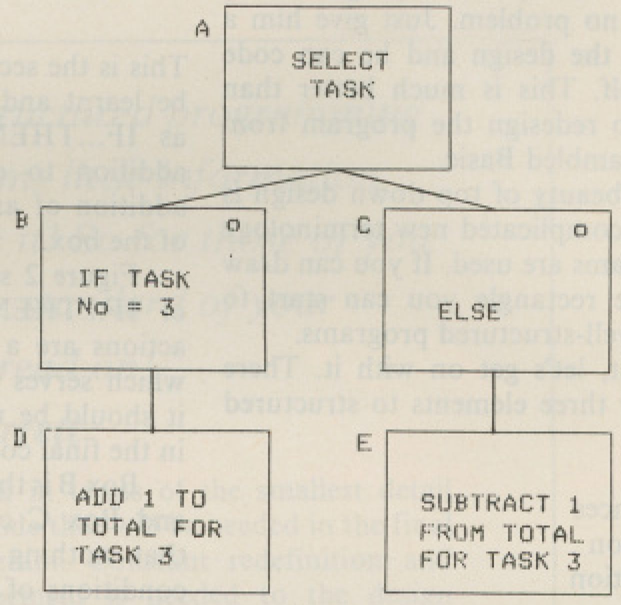


Figure 3 – selection with alternative action

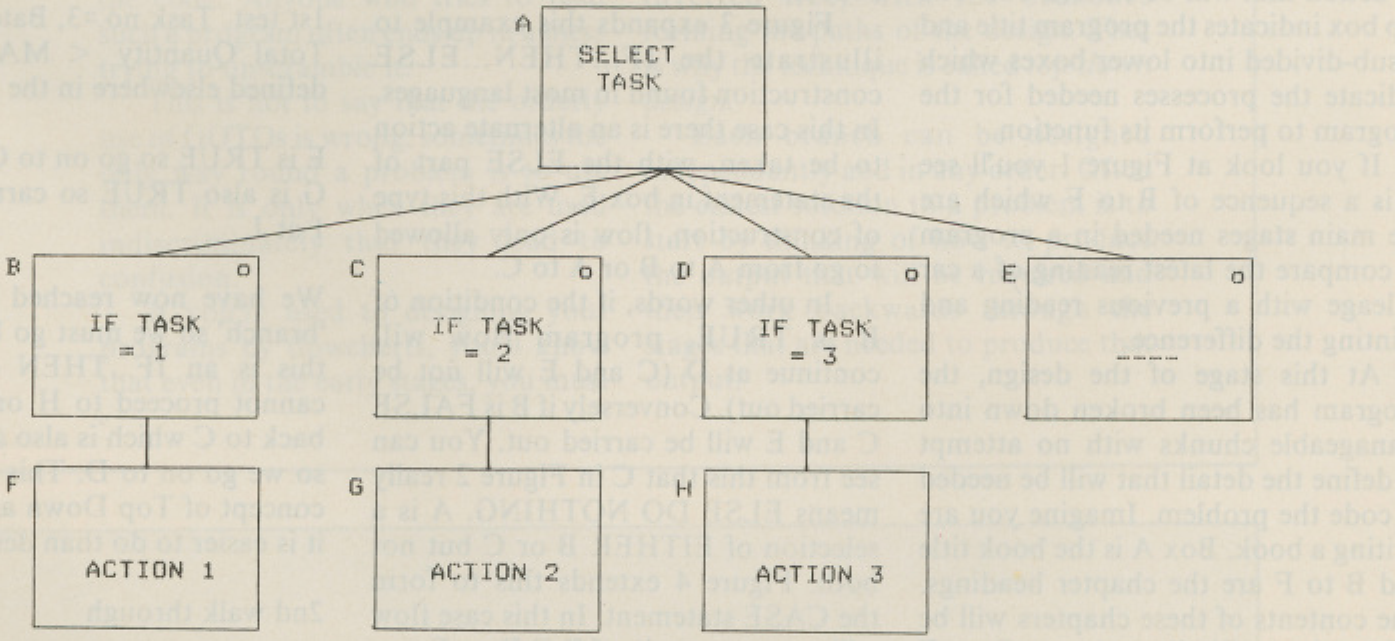


Figure 4 – structure of CASE statement

Quantity < Max.

E is TRUE so go on to G
G is FALSE so go to H
H is TRUE so go to K

K is TRUE so do Action 2 at M then Action 3 at N. I have added a sequence to show the possibilities of design. J,M and N would be sub-divided in a real program.

3rd walk

Task no.=3, Batch Total=260, Total Quantity < Max.

E is TRUE so go to G
G is FALSE so go to H
H is FALSE so go to I
I has no action so carry on at D.

4th walk

Task no.=4, Batch Total=350, Total Quantity > Max.

E is FALSE so go to F
No action at F so carry on at D.

Get the idea now? But was it the result you expected? If it was you have succeeded in your design. If it wasn't then go back and re-design the bits that didn't work. A lot of refining is often needed at this stage, but the more time spent getting the design right leads to less frustration when you run your masterpiece and find that it doesn't work as you expected, or, even worse, doesn't work at all.

Repetition

On to the last of the three elements. That of repetition, also known as iteration. You've all used the familiar FOR...NEXT construction of Basic, and possibly the allied REPEAT....UNTIL or WHILE....DO of more advanced languages. These are all forms of repetition. Repetition is merely the forming of a loop to do the same action many times over.

Once again we will use the familiar rectangle but this time add an asterisk (*) in the upper right of the box that is used as the control element.

I'm afraid space doesn't allow us to run the whole article, and so look out for the second installment in the near future.

See listings on page 73

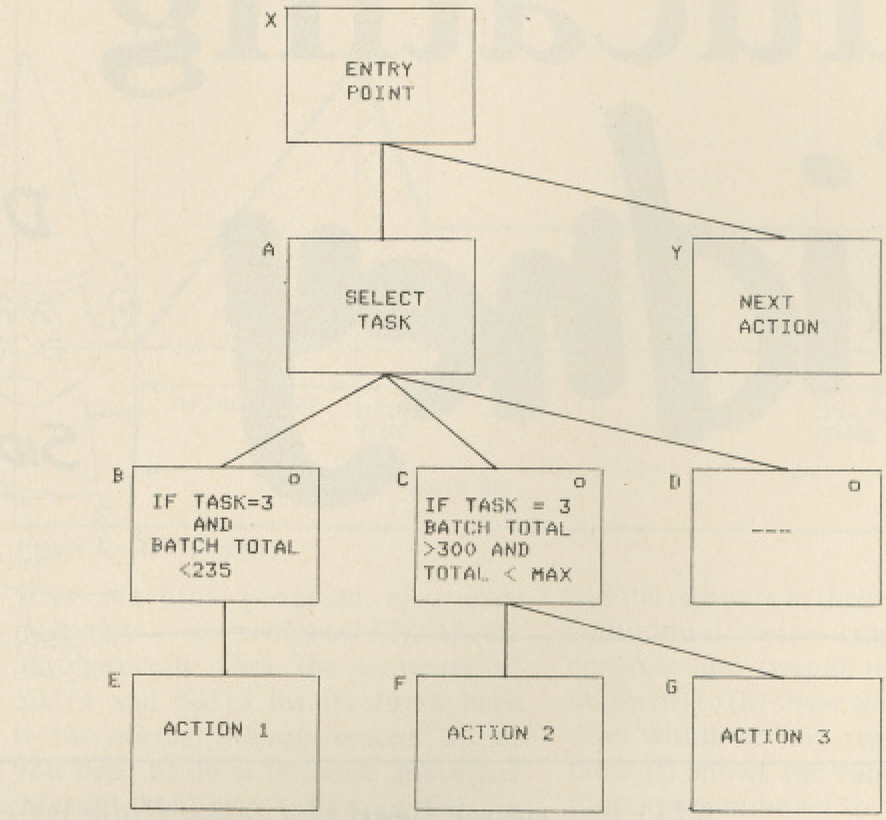


Figure 5 – selection with multiple conditions

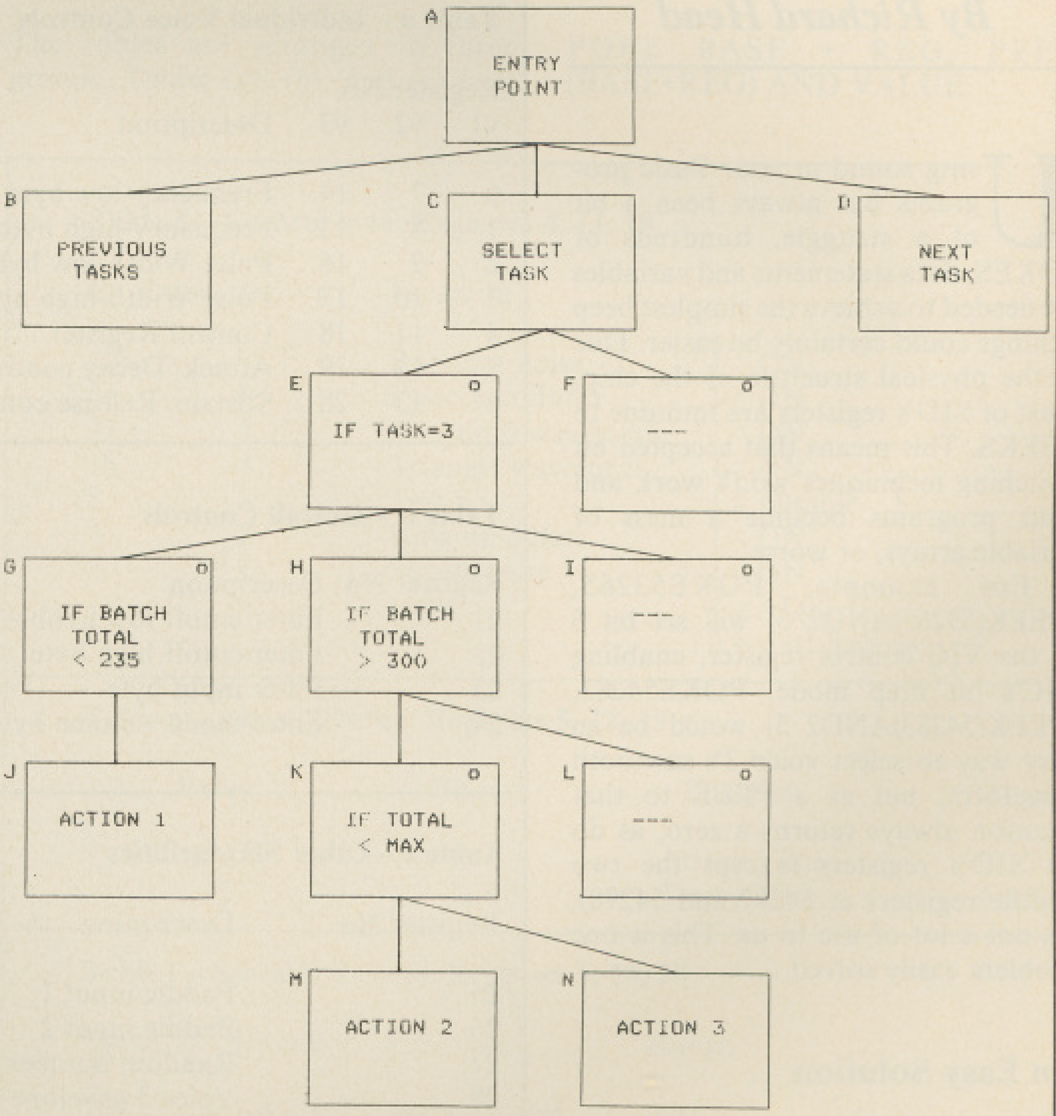
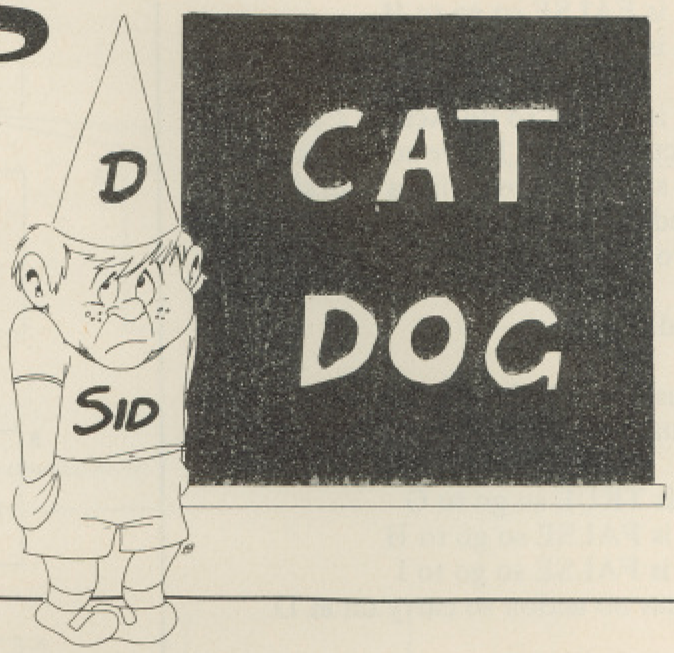


Figure 6 – alternate structure for figure 5

Educating Sidney



SID has never been the friendliest of characters, teaching him to be a little more sociable can't hurt.....

By Richard Head

Using sound in your Basic programs has always been a bit of a struggle, hundreds of POKES, data statements and variables are needed to achieve the simplest beep – things could certainly be easier. Due to the physical structure of the chip, most of SID's registers are immune to PEEKS. This means that accepted bit switching techniques won't work and your programs become a mess of variable arrays, or worse!

For example, POKE53265,(PEEK(53265)AND2^5) will set bit 5 of the VIC control register, enabling VIC's bit map mode. POKE54283,(PEEK(54283)AND2 5) would be an easy way to select voice 2's sawtooth waveform, but as a PEEK to that location always returns a zero, as do all SID's registers (except the two paddle registers at 54297 and 54298), it's not a lot of use to us. This is one problem easily solved.....

An Easy Solution

Enter SIDREAD. When initialised, this short machine code program

Table a – Individual Voice Controls

Register No. v1	v2	v3	Description
0	7	14	Frequency-low byte
1	8	15	Frequency-high byte
2	9	16	Pulse Width-low byte
3	10	17	Pulse Width-high nybble (0-15)
4	11	18	Control Register
5	12	19	Attack/Decay control
6	13	20	Sustain/Release control

Table b – Overall Controls

Register No	Description
21	Filter cutoff low nybble (0-7)
22	Filter cutoff high byte
23	Filter input byte
24	Filter mode/volume byte

Table c – Other SID facilities

Register No	Description
25	Paddle input 1
26	Paddle input 2
27	Random number generator
28	Voice 3 envelope output

wedges itself into the 64's interrupt system. All it does is set up a dummy SID in RAM, copying its contents into the real SID every 60th of a second. Once the program is installed, any value poked into the RAM table is automatically copied into its corresponding SID register. Thus you can PEEK the RAM table, perform your maths on the value, and POKE it back again. The earlier example will now work, once the address has been changed to point to the RAM table.

The code was originally written to sit at $C000 (49152), and load in direct from disk, but to make things easier still, I've produced a Basic generator program that will install a working version of the code anywhere in RAM, and save a copy to disk. SIDREAD is the program generator, and before you run it, make sure you set the base address in line 60 (it defaults to 49152 ($C000). Beware though, if you try to put the code under the ROM, or any reserved RAM area (zero page, screen RAM etc), it won't work.

When you Run the generator, the start and table address are displayed and it's probably worthwhile writing them down. Disk users get the option here of saving a working copy of the code to disk, just follow the screen prompts. To load the code back again, use LOAD"filename",8,1 .

With the code in place, SYS(base address) will clear SID and initialise the table. Any poke to the table will automatically be copied to the corresponding SID register. Should you happen to hit RUNSTOP/RESTORE, you will need to re-initialise the table before using it again, use either SYS(base address) or SYS(base address + 10). The latter will re-initialise without clearing SID.

Getting it all in

SIDTEST is a demo program, load and run it, and nothing happens! This is because the program relies on PEEKing SID. Next, load and run the generator program, and re-load SID TEST. Now change line 10 to read SID=(base+49) and RUN (if SIDREAD was Run unaltered, (base+49) should equal 49201). If all has gone well, you'll know about it!

Once you've created a working copy of the code at a suitable address, you're not restricted to using it in Basic programs; make use of it in your machine code programs as well! If your machine program also uses interrupts – no problem. SIDREAD automatically uses the contents of $0314 and $0315 for its Jump back to the normal interupt system, so all you need to do is initialise any other interupt routines before initialising SIDREAD.

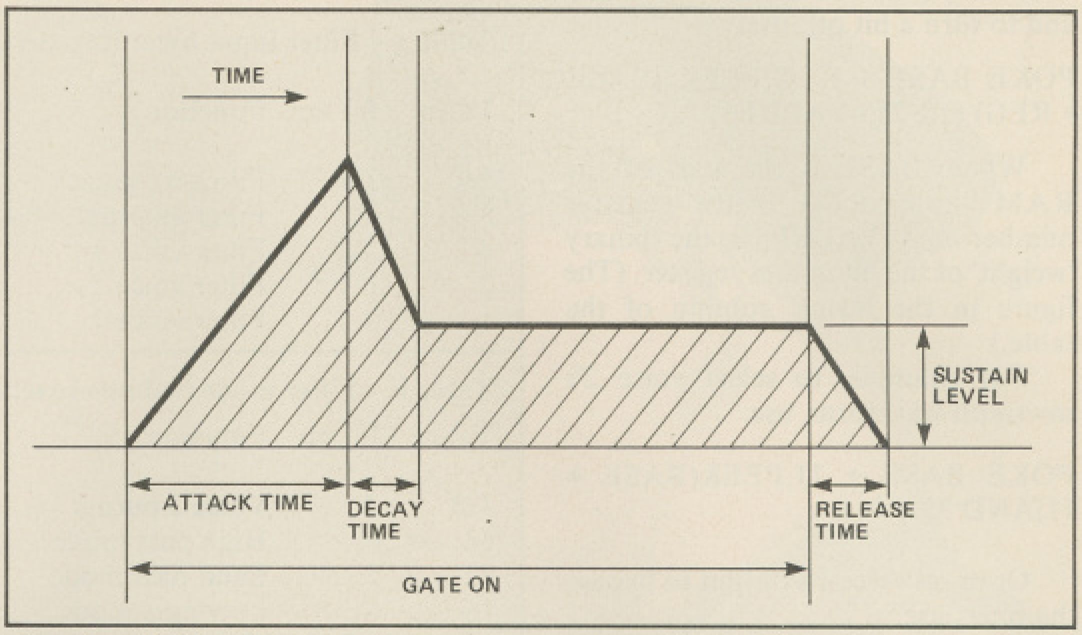


Figure 1 - ADSR cycle

Using the Tables

The tables are arranged in three groups. Tables (a), (b), and (c) map out the registers in their various groups (individual voice controls, overall controls and output registers), while tables (d) to (h) show what each section does within certain registers. Finally, table (i) shows the ranges that SID's functions can be set to.

To set any single bit in a control register to a '1', use this simple formula (in conjunction with the SIDREAD program):

POKE BASE + REG, PEEK (BASE+REG) AND VALUE

Table d – Control byte: Registers 4, 11, 18

Bit Value	Bit no	Function	
128	7	Random noise	: 1 = ON
64	6	Pulse waveform	: 1 = ON
32	5	Sawtooth waveform	: 1 = ON
16	4	Triangle waveform	: 1 = ON
8	3	Test bit	: 1 = Disable
4	2	Ring modulate	: 1 = ON
2	1	Synchronise	: 1 = ON
1	0	Gate	: 1 = ADS, : 0 = start Release

Table e – Attack/Decay: Registers 5, 12, 19

Value	Bit nos	Function	Range
240	7-4	Attack Time	0-15
15	3-0	Decay Time	0-15

Table f – Sustain/Release: Registers 6, 13, 20

Value	Bit nos	Function	Range
240	7-4	Sustain Level	0-15
15	3-0	Release Time,	0-15

and to turn a bit off, use:

POKE BASE + REG,PEEK (BASE + REG) OR 255-VALUE

Where BASE is the start of the RAM table, REG is the register number and VALUE is the binary 'weight' of the bit in that register. (The figure in the 'value' column of the table.)

For example: to select voice 2's sawtooth waveform, use:

POKE BASE + 11,PEEK(BASE + 11)AND 32

Or to set voice 1's output to bypass the filter, use:

POKE BASE+23, PEEK (BASE+23) OR 255-1

Figure 1 is a graph of volume against time, showing how SID's envelope generator works. The volume or 'amplitude' envelope of a sound is one of the primary factors deciding what the sound will 'sound' like. Try experimenting with different values and note the differences.

See listings on page 73

Table g – **Filter input byte: Register 23**

Value	Bit no	Function	
240	7-4	Filter resonance	: 0-15
8	3	Filter external	: 1 = yes
4	2	Filter voice 3	: 1 = yes
2	1	Filter voice 2	: 1 = yes
1	0	Filter voice 1	: 1 = yes

Table h – **Filter mode/volume byte: Register 24**

Value	Bit no	Function	
128	7	Voice 3 output	: 1 = off
64	6	High pass mode	: 1 = on
32	5	Band pass mode	: 1 = on
16	4	Low pass mode	: 1 = on
15	3-0	Master volume	: 0-15

Table i – SID Function Ranges

Function:	Range:	Value:
Oscilator frequency	0-4KHz	0-65535
Attack time	2ms-8s	0-15
Decay time	6ms-24s	0-15
Sustain level	0-peak volume	0-15
Release time	6ms-24s	0-15
Pulse width	0-100%	0-4095
Filter range	30Hz-12KHz	0-2047



The Normandy landings produced many heroic contests, however it didn't provide a direct confrontation between Patton (who arrived one week after the landings) and Rommel (who was injured one week before). Electronic Arts, not letting facts get in the way of a good simulation, has produced a game that matches these two in a head to head fight.

The game was written by Chris Crawford, who wrote the all time classic Eastern Front, and provides the same simple but realistic game mechanics and a tough computer opponent.

Unlike, other wargames where the object is to defeat the enemy forces, Patton vs Rommel, is a battle for territory in the shape of towns. The allied forces have just a few days to break through the German lines and capture as many towns as possible by moving through them. The German player must halt their progress and maintain their thin line of defence for as long as possible.

The state of the game, which is played in turns, is represented as a victory point total which will begin and stay negative until the allies take towns, when it will gradually increase until a positive score announces an allied victory. A German player must keep the score negative to win the game either by halting the allied advance or by retaking captured towns.

The screen display is entirely black and white, which is never explained except it may be to mirror the Macintosh. Though why ignoring the C64s excellent graphic capabilities is beyond me.

Most of the screen is filled by a fraction of the battle map and can be scrolled across it by moving a window over the map icon. The dark areas represent high ground and slows down progress that enters it. Conversely, the white roads connecting the towns form the quickest way to travel.

There are just two types of units in the game infantry and tanks although these symbols actually represent more accurate forces, such as supporting artillery which are displayed whenever a unit is selected. By clicking other icons the units can be displayed instead as arrows showing which way they are facing, circles representing the total and actual strengths and by dots or widening cracks showing the damage they have sustained. A badly cracked unit could run if it met a determined and even weaker opponent.

To issue an order to a unit is a simple case of selecting (clicking) it and then pointing to where you want it to go. In the basic game it will try and move there as quickly as it can but will stop and fight any enemy units it goes near. In the intermediate level you must give more specific orders by clicking icons that command a unit to turn clockwise or anti-clockwise and move forward in one of seven modes which range from the rapid movement, but open to attack road to the artillery assault of static attack mode.

Whatever your orders the computer will then carry them out before either Patton or Rommel, and where appropriate will comment on your performance and may even pass on the odd tip.

Once you have broken through or held the lines of defence several times you may be looking for the new challenge that's ready and waiting, and allows you to edit the game to produce a much tougher Expert level game.

Patton vs Rommel has a few quirks such as the mono display, but is well thought out and will challenge even experienced computer generals.

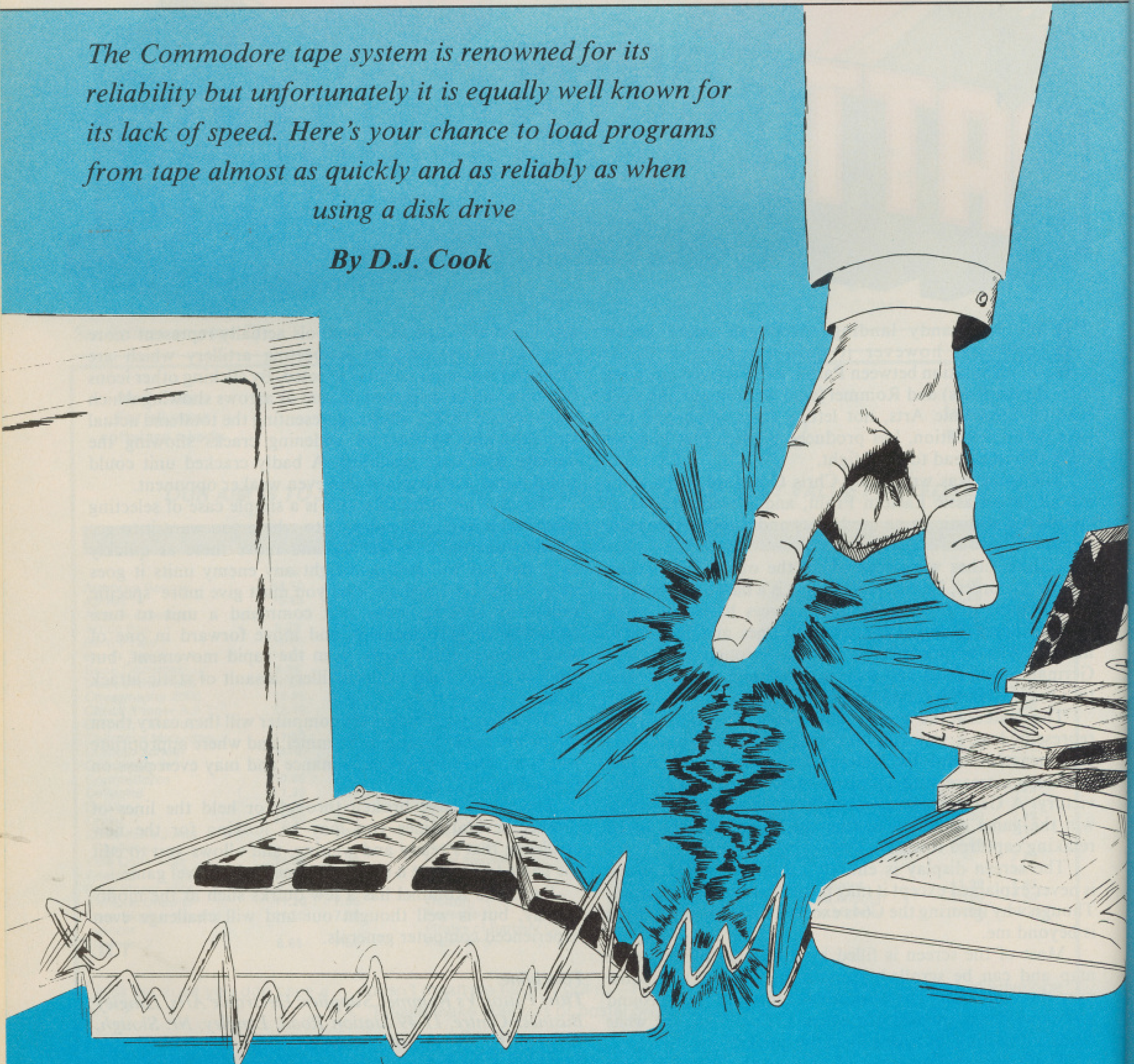
T.H.

Touchline:
Title: *Patton Vs Rommel.* ***Supplier:*** *Electronic Arts, Langley Business Centre, 11-49 Station Road, Langley, Nr. Slough, Berks SL3 7YN.* ***Tel:*** *0753 49442.*
Machine: *C64.* ***Price:*** *£14.95 disk only.*

TAPE ORGANISER

The Commodore tape system is renowned for its reliability but unfortunately it is equally well known for its lack of speed. Here's your chance to load programs from tape almost as quickly and as reliably as when using a disk drive

By D.J. Cook



The program uses a directory program which is saved as the first program on a tape and stores all the names of all the programs on the tape presenting them as a menu. Once a program is chosen the tape is wound on at fast speed to the start of that program, as this wind-on is controlled entirely by the directory program there is no need to sit watching the tape counter.

The program can then be loaded using the fast loader which is automatically installed by the directory.

Creating The Program

First of all type in program 1. This program contains the machine code fast load routines. When the program is correct SAVE a copy for later use and then run the program.

Having run program 1 you should delete the program using NEW and then type in program 2. Do not run the program at this stage or you may cause the computer to crash with the SYS command. SAVE a copy of the program.

When you are satisfied that the program is correct you should put a blank tape into the recorder with the junction of the leader tape and the magnetic recording tape positioned on the felt pad of the cassette (accurate positioning of the tape each time the directory is altered and resaved means the directory never overwrites other programs on the tape).

With the tape in position type in SYS 49887. This calls a machine code routine to transfer the fast loader to the end of the BASIC program and then saves both the BASIC and machine code as a single composite program. The composite program can be loaded and saved in exactly the same way as a normal BASIC program but once the machine code is added the BASIC program cannot be edited. If there are any errors in the program they must be corrected before calling SYS49887.

Saving a Program Using the Directory

To save a program from an ordinary tape onto a directory controlled tape you should first load the directory from the beginning of the cassette and overwrite the appropriate DATA statament with the name of your program. Take care not to alter the length of the DATA statement and check that PEEK (3072) returns the number 169. Once the DATA statement is correctly altered you should then resave the altered directory in the same position on the tape as the original (by positioning the junction between the leader tape and the magnetic tape on the felt pad of the cassette). Do not press stop, or rewind the tape. Run the program and follow the screen prompts. The tape will wind on automatically to the start of the sector in which the program is going to be stored. Remove the tape without rewinding and keep to one side. Now load the program you want to save and when correctly loaded replace the directory tape in the recorder and fast save the program using £S "name of program". The program is now saved on the appropriate sector and can be directly accessed by the directory.

Loading Programs Using the Directory

This is the easy part. Load and run the directory. Follow all the screen prompts (especially the one that says stop the tape!). Use £L to load your program and press the space bar or Commodore key once the FOUND .... appears and that's all there is to it. This program is very efficient at finding and loading programs and is the most invaluable program I have ever used.

I never use cassettes larger than C20 as I find longer tapes take more time to wind on to programs and they stretch and break more easily. One side of a C20 can store up to ten 20K programs which I find is enough on one tape though the directory does have sufficient DATA statements for up to 20 programs if you want to use longer tapes.

The sectors defined by this program are large enough to store programs up to 20K, if a longer program is saved it will occupy more than one sector so be careful when using long programs not to overwrite any programs already stored on the tape. If most of your programs are longer than 20K then the sector size can be increased by increasing the 250 in line 130 to suit your own requirements, or if your programs are much shorter you can save time and tape by decreasing the 250 appropriately.

How the Directory Works

The tape motor can be controlled by POKEing registers 1 and 192 with the appropriate numbers (POKE 1,39 stops the tape and POKE1,7 starts it again) as in lines 110 to 140. Unfortunately the operating system constantly checks to see if a key is depressed on the datasette and resets both registers appropriately. The way I have got round this is to have a tight loop constantly POKEing the registers until the space bar is pressed. This allows the datasette to be changed from PLAY to FASTFORWARD etc without the tape motor starting up. When the space bar is pressed the tape motor is started (by POKE 1,7) and the tape is wound on at fast forward speed to the beginning of the chosen sector.

The timing of the wind-on uses TI, the internal clock of the 64 which is incremented every 1/60th of a second. How long to wind on, and therefore how far along the tape the sector is, is determined by the number of the chosen program and a constant (250 in this case). This divides the tape into sectors of approximately 120cm length which is about 50 secs of recording time or 4.2secs at fast forward speed. Each sector can hold about 20K of program using the fast load routines or slightly less than 2K using a normal save.

The machine code fastload is transferred into memory from 49152 to 50176. This area is not normally used by BASIC and should not cause problems with BASIC programs but the fastload will be incompatible with any machine code routines which use this area of memory. The program alters the character dispatch vector ($308 & 309) to allow the £L, £S and £V commands to be implemented. If you want to disable the fast load routines then SYS58451 will restore the normal vectors while SYS49215 will reinstate the fast load.

The directory program can be used without the fast load using the normal save and load routines but the 250 in line 130 needs to be greatly increased to accomodate the greater tape length needed by the slow save. If the machine code is omitted then the SYS command mush also be omitted or the computer will crash.

See listings on page 73



Here's your chance to become a junior wizard, capable of wielding limited powers who sets out to become the chief wizard by beating the best seven magicians in the land.

Armed with only four spells in your spellbook you stride out into the wilderness which scrolls inside an unusual circular window which is flanked on either side by the spellbook, image and physical, spiritual and mental levels of you and your opponent.

The task ahead is split into three levels that must be completed in order. First of all you must explore seven towns that are spread across the land and do battle with six monsters. Defeating these monsters will grant you access to the treasure they guard which can be traded with the relevant town for food to top up your three energy bars. Once you have killed all six monsters and have delivered the treasure to all six towns the seventh leads to level 2.

Level 2 provides a test to see whether a wizard is strong enough to challenge the seven master magicians. The contest comprises 30 monsters each with unique strengths and abilities. Three of these possess magical artifacts that must be collected if the player is to gain access to level 3 and a duel to the death with the seven wizards.

To complete the game the player must fight and defeat each of the master magicians, beginning with the Wolf Lord and Bear Lord and ending with a final battle with the Dragon Lord.

The key to success in your quest lies in the spells that you wield from the four that you can select at the beginning of the game, to the others that you earn by defeating the monsters in level 2. Any spells that you gain must be stored on an empty page in your spell book, so there are times in the game when you will have to face the difficult decision as to which spell to erase from your book.

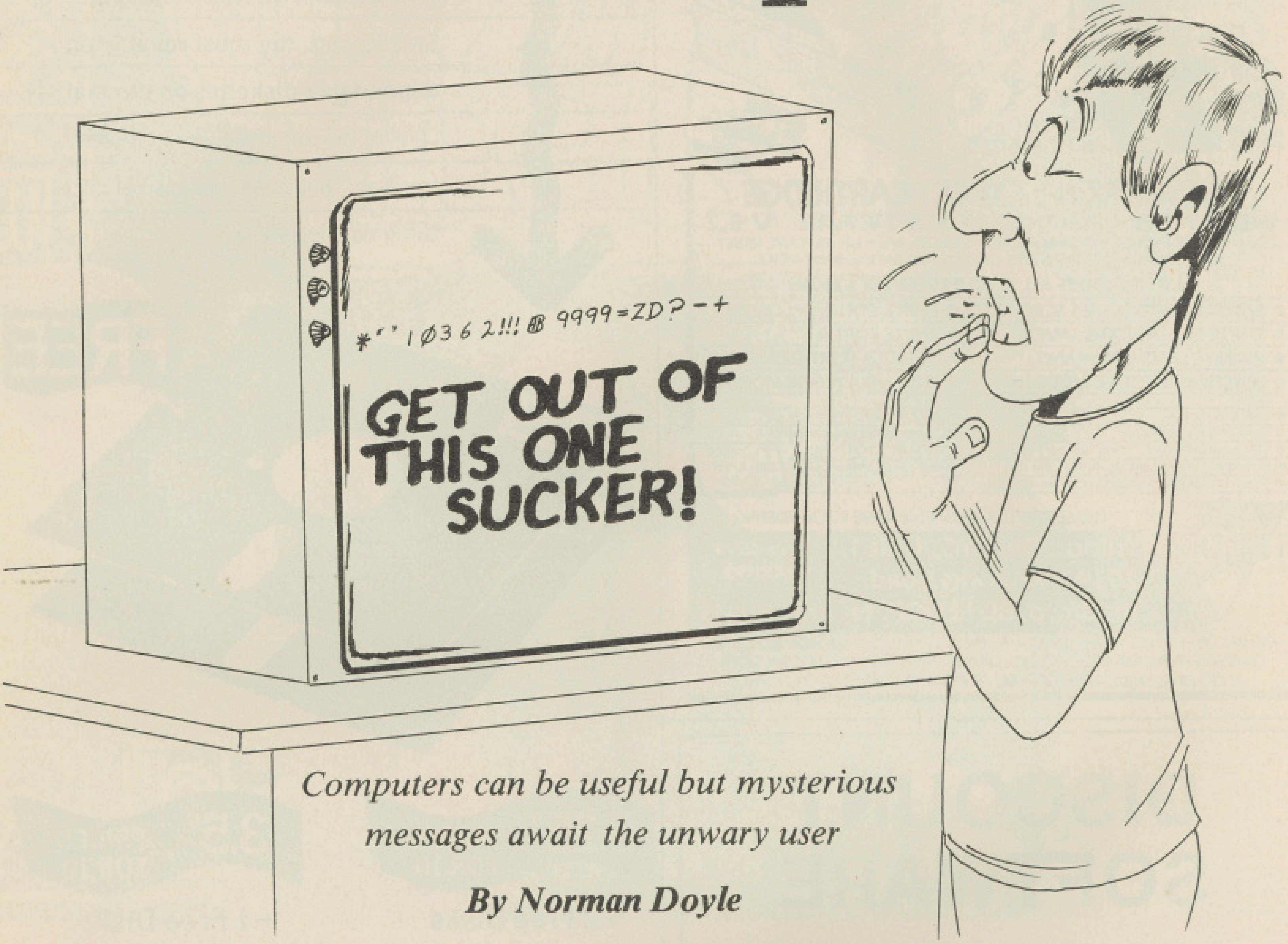
The spells are split into three groups and represent the magic drawn from physical, spiritual and mental power. Physical spells include the more usual fireballs, walls of ice, magic missiles and rock showers associated with fantasy games. However in Wizard Warz you can wield the Fear, Evil eye, far Vision and protection from evil spiritual spells and the mental trickery of Forget, Invisible and Mind rack.

The result is a mixture of arcade action and roleplaying that creates one of the first true arcade adventures. **T.H.**

Touchline:

Title: *Wizard Warz.* ***Supplier:*** *GO! (US Gold) Units 2/3 Holford Way, Holford, Birmingham B6 7AX.* ***Tel:*** *021-356 3388.* ***Machine:*** *C64/128* ***Price:*** *£9.99 (Ca) £11.99 (Disk).*

First Steps



Computers can be useful but mysterious messages await the unwary user

By Norman Doyle

Cast your mind back to the early days of computing. A computer utopia was dawning, the ZX 81 had matured into the Spectrum and the VIC-20 was being superceded by the Commodore 64. Magazines were full of articles relating to the revolutionary benefits of the new technology. Home accounts would be held on computer, databases would hold all the names, addresses and birthdates of friends and relatives, and the computer would even control the improved household gadgets which would result. To brutally paraphrase Plato, the reality has proved to be less than the dream.

Small businesses could certainly benefit through the use of the C128 and many would even find the C64 and disk drive beneficial but the problem here is one of credibility. Amstrad's Alan Sugar observed recently that business people prefer to use equipment which is not identified with the domestic market. Bigotry such as this probably explains why the Commodore PET was so successful with large and small businesses but, in the shadow of the PC, the C128 has not prospered.

Application Error

Computopia was not achieved for one main reason – in most cases the tasks are not complicated enough to merit the time and trouble of loading and using the software. Until hard disks become cheaper or an even better instant retrieval system is devised, home acounts will be worked out on cornflake packets, hi-tech database will still be called Filofax and home gadgets will retain an unfathomable mind of their own. The only area where a computer has proved beneficial is in the realm of wordprocessing and leisure pursuits.

One of the most rewarding pastimes is programming – it stretches the mind and usually rewards the programmer with something that can be used over and over again. One of the less pleasurable aspects is debugging. Everyone would like to create something that works first time but that isn't always possible.

Whatever the make of the computer, one thing they all have in common is the fearsome error message generator. The Commodore 64 has 33 of these messages and the C128 has 44, as shown in the panel. Over the next few issues *First Steps* will look at each message in turn and consider the reasons why these occur and how to find, correct and avoid them.

Too Many Files

Occurence: rare

When this error is reported the program has attempted to open more than ten files. The simple solution is to check that all the files are necessary and prune out those which are not needed or redundant.

If such an inordinate range of files is necessary, it is unlikely that they will be receiving information simultaneously. In multiple file programs, it is always good practice to open a file, read or write the information necessary and then close it again immediately.

File Open

Occurence: program error
When opening a file there are three values following the OPEN command. The first is the file number which can, with few exceptions, be any number between 1 and 255. The second value is the device number and the third is the secondary address which determines the kind of operation that is to take place.

The file number is the one used when communicating with a device using file handling commands such as PRINT#, INPUT# and GET#. This error means that an open file is already using the value. If you like, the file number is a connection from the computer to one peripheral device. Although several of these lines can be attached to the same device, it is not possible to connect more than one peripheral on a single line.

The solution to this problem is to allocate a different number to the file. With 255 numbers to choose from this shouldn't be difficult because only ten files can be open at any one time (see too many files).

File Not Open

Occurence: program error
This also relates to the file number and occurs when a file handling command uses a value which has not yet been allocated to any device. The answer to this is simple, either open a new file or find a file which has already been opened and allocate the value to the handling command.

File Not Found

Occurence: user error
This is principally a disk error and simply means that a file for loading or verifying has not been found on the disk currently in the drive. This can result when the wrong disk is in the drive, if the directory has been corrupted in some way, or through typing errors.

The only way around the corrupted disk error is to shrug your shoulders and promise to be more careful in future, format a new disk and start again. Otherwise the answer is to check the directory and then insert the correct disk or correct the spelling of the file name.

Be careful and this error should never occur.

Device Not Present

Occurence: user error
This is displayed as a file command issued to a peripheral which is not connected or is turned off. Care should be taken when this occurs. If the device is connected it can usually be switched on with no problem. If it has not been connected the only safe solution is to switch the computer off and start again after connecting the device.

If the device is the cassette recorder or disk drive and a program is in memory, which will be lost if the computer is switched off, drastic measures must be taken. Ensure that the device is switched off and that any other peripherals connected through it are also off. Next check that the plug is correctly positioned for insertion and then confidently push the device connector into the relevant computer port. DON'T wiggle the plug about. Finally switch the device on.

This procedure is contrary to any safety regulations and should only be used as an emergency measure. Lack of care will result, at best, in the computer doing a warm start or reset – at worst you could damage the input/output chips inside the computer, peripheral or both!

Cartridges and user port connections should ONLY be done with the computer turned off. It is very easy to misalign these connections and short across terminals which should never ever meet. YC

Table 1 : Commodore Error Messages

Peripheral Errors

TOO MANY FILES
FILE OPEN
FILE NOT FOUND
DEVICE NOT PRESENT
NOT INPUT FILE
NOT OUTPUT FILE
BAD DATA
MISSING FILE NAME
ILLEGAL DEVICE NUMBER
MISSING FILE NAME
LOAD
VERIFY
FILE DATA

Mathematical Errors

ILLEGAL QUANTITY
DIVISION BY ZERO
FORMULA TOO COMPLEX
UNDEF'D FUNCTION
OVERFLOW

Input Errors

STRING TOO LONG
TYPE MISMATCH
EXTRA IGNORED
REDO FROM START
BAD SUBSCRIPT

Operational Errors

NEXT WITHOUT FOR
RETURN WITHOUT GOSUB
OUT OF DATA
OUT OF MEMORY
UNDEF'D STATEMENT
REDIM'D ARRAY
ILLEGAL DIRECT
BREAK
CAN'T CONTINUE
SYNTAX

Additional C128/Plus4 Errors

CAN'T RESUME
LOOP NOT FOUND
LOOP WITHOUT DO
DIRECT MODE ONLY
NO GRAPHICS AREA
BAD DISK

Extra C128 Errors

BEND NOT FOUND
LINE # TOO LARGE
UNRESOLVED REFERENCE
UNIMPLEMENTED COMMAND
FILE READ

Infiltrator II



The over-the-top hero Captain Johnny"Jimbo-Baby" McGibbetts is back for another series of adventures in Infiltrator II.

The world class printer, TV and film mega star, consultant to every free world leader and all round nice guy has been called back into action to once again thwart the Mad Leader's evil schemes. The call came just in time as you were sliding from the front page 23! Not exactly the ideal situation for a hero who craves the limelight. Now the Mad Leader has reorganised his forces and you're back in business in another trio of impossible missions.

Each mission consists of three parts. First you must pilot your Whizzbang Enterprises Gizmo DHX-Attack helicopter through enemy lines, infiltrate an enemy base to complete a ground mission and then fly back to homebase and rapturous applause. Fly three such missions and you're once again the hero you think you are.

In the first mission you must discover the Mad Leader's chemical labs and neutralise a vast of deadly nerve gas, then return in part two to discover and deactivate his missiles and cause a meltdown in the Leader's nuclear reactor before completing the game by tracking down the madman himself and by installing a brain plant which turns into a valuable member of society!

Flying the Gizmo to the target requires a combination of skill to keep in the air and to make sure you arrive at base with enough fuel to get home, strategy to plan and discover your route and skill deciding your reactions to other aircraft that patrol the skies.

Through a series of passwords that aircraft exchange, it is possible to decide who is friend and foe. However, you are the Infiltrator which happens to be your password for friendly forces, but you can fool the enemy with the password ID Overlord. If you give the wrong response then you'll need all your wits as well as your Gizmo's guns, rockets, flares and chaff launchers to survive the battle.

At the base, the screen display and game changes from a flight simulator that incorporates thumbovision to a tense ground mission. Whichever mission you are on, you will be up against overhwelming odds and so you will have to use cunning and stealth to gain entry rather than attempt a doomed assault. Armed with joystick and selected false papers you should be able to get past most guards, however our hero has a sleeping gas cannister up his sleeve in case the papers don't impress and a stock of grenades if things get really tough.

Inside the buildings the display changes once again as McGibbets searches through the rooms for more weapons, a change of uniform, security passes and the key to his mission either the vat of chemical gas, missiles or the Mad Leader himself.

The game is both challenging and enjoyable as you relish the chance to be a hero that's completely over the top, and to add to the value the original Infiltrator is included in the game pack. Two games for the price of one can't be bad!

T.H.

Touchline:
Title: *Infiltrator II.* ***Supplier:*** *US Gold, Units 2/3 Holford Way, Holford, Birmingham B6 7AX.* ***Tel:*** *021-356 3388.* ***Machine:*** *C64/128.* ***Price:*** *£9.99 (Ca) £14.99 (Disk).*

SPLIT RATE

Save on phone bills with this intelligent modem software

With the prices of hardware constantly falling, bulletin boards are using intelligent modem equipment that accept a variety of incoming baud rates. The caller wants to use as high speed as possible to minimise both his waiting time and telephone bill. Cheap RS232 type modems can often cope with both 300/300 and 1200/75 baud and unless you are planning to upload software, 1200/75 is by far the best to use – after all, most of your time is spent reading rather than writing.

A scrolling terminal emulator is very easy to program if you restrict yourself to 300/300 since the Commodore RS232 implementation can cope with this directly. There is even a perfectly useable BASIC version in the Programmers Reference Guide. However, to allow the computer to send at a different rate to the one at which it is receiving requires a little more work. 1200 baud is also too fast for BASIC and machine code needs to be used to make sure that the receiver buffer does not overflow.

The real benefit of the faster reception is when it is used to download software from a bulletin board. However, there is always the problem of telephone line noise, and so, some form of error checking is needed. This program uses the very common xmodem protocol.

How it Works

The program consists of a BASIC section that does all the setting up required. There is no rush for this. Terminal emulation and the xmodem part are in the machine code section.

Terminal emulation consists of sending the character codes of the keys pressed to the RS232 output and printing the received character codes onto the screen. Bulletin boards work in proper ASCII which has the upper and lower case letters reversed, compared with Commodore. This conversion needs to be performed for both transmitting and receiving. Any control codes may be sent to the RS232 output, but they are filtered out of the input to prevent misreceived characters effecting the display. Standard KERNAL routines have been rewritten to allow an independent baud rate.

The xmodem protocol consists of sending or receiving 132 byte packets. Each packet consists of SOH (ASCII 1) followed by the packet number and the 2's complement of the packet number. Then, there are 128 bytes of data and finally a single byte checksum calculated by taking the least significant byte from the sum of the data bytes. Transfer is initiated by the receiving computer sending NAK (ASCII 21).

The sending computer then sends out the first 132 byte packet. If the packet number and the checksum are correct, the receiving computer will them send ACK (ASCII 6) and the sender will send the next packet, but if either of these values are wrong, then NAK is sent to tell the sender to repeat that packet. The final packet has its data padded out to 128 bytes by control-Z (ASCII 26) characters, and when the sending computer is asked for another packet, it sends EOT (ASCII 4) instead of SOH and waits for a reply of ACK before stopping.

This may all seem a little long-winded, but it does mean that any transmission error should be detected, and the whole file should be transferred correctly. A mistake of a single byte can mean that a program will not work.

Getting it all in

There are two versions of the program; one for the C64 and one for the C128. These are both BASIC programs which poke in the machine code program each time the progam is run. The machine code section has built in error checking and the BASIC section can be modified as required. Simply type it in. The C128 version works in both 40 and 80 column modes and automatically switches to fast mode where possible.

BAUD TERMINAL

By William Sellers

In Use

When the program runs without any errors, you will be confronted by a menu. The baud rate and protocol options are self evident. If neither is used, then the RS232 port is set to 300/300 baud, 8 data bits, 1 stop bit and no parity. This can be changed if required.

Online opens the RS232 channel and enters the terminal mode. This will switch most modems online, but you may have to do this manually. Normally, you need to dial the required telephone number, wait until you hear the computer answer tones and then switch the modem to online. You may also need to hit return a few times to get the other computer to acknowledge you. To get back to the main menu, press Commodore-Q.

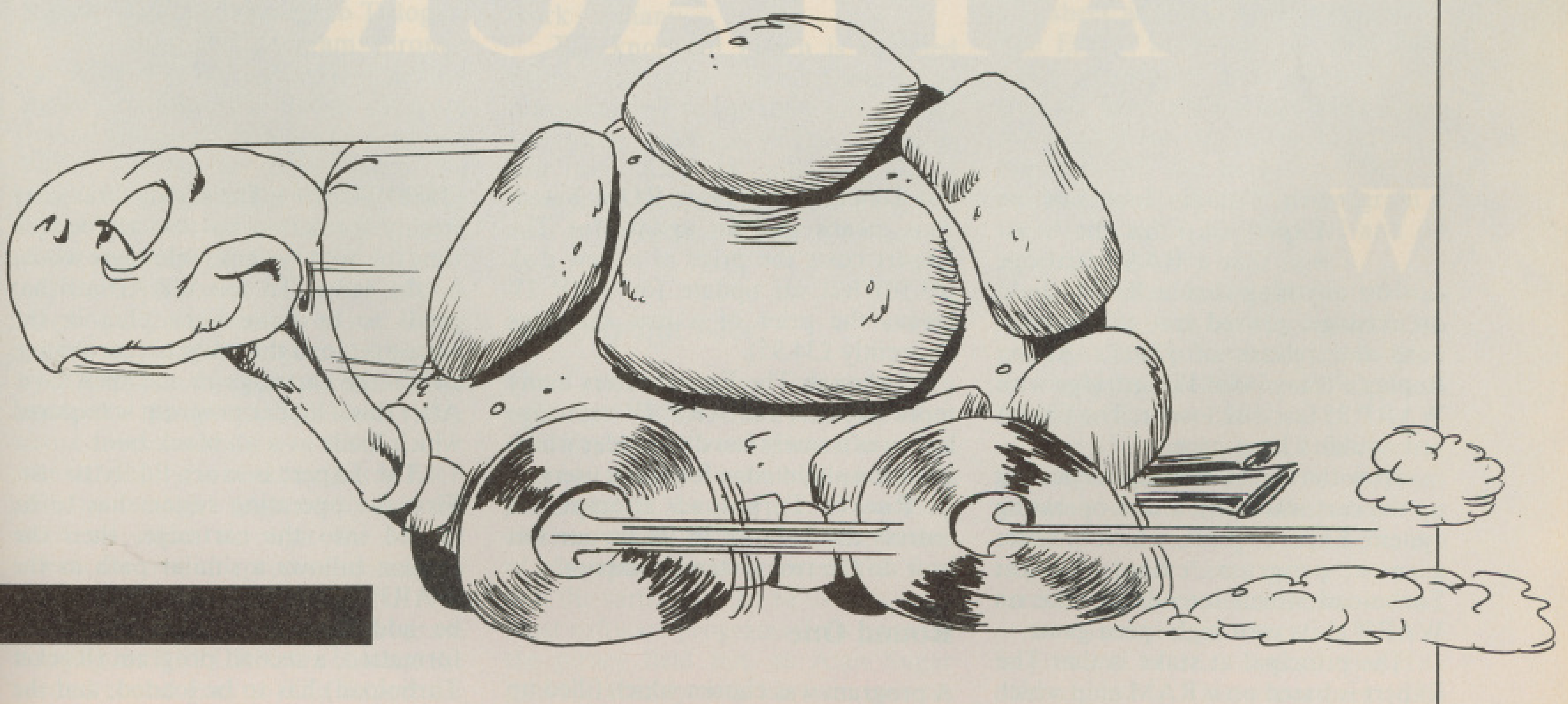
Xmodem send and receive are (surprisingly) for sending and receiving files with the xmodem protocol. To use these, access the bulletin board with the online option and activate file transfer at that end. Then press Commodore-Q to return to the menu and select the send or receive option as required. You will then be prompted for a filename and the display will show the status of the file transfer. When this has finished, you will be returned to the main menu. To abort at any stage, press Commodore-Q. Xmodem on the host is often cancelled with Control-X. Note: xmodem will only work with 8 data bit protocols.

ASCII/Pet conversion is for converting between the two character coding systems. Conversion from ASCII to Pet codes also removes any line feeds from the file since these are usually unwanted. Other control codes are left unaltered.

Other Uses

Xmodem is often used for file transfer along RS232 lines between computers so this program is quite handy for getting your Commodore to talk to any other computers you may own – assuming you have suitable software for them.

See listings on page 73





Can the new Expert cartridge disk system broadside WARP 25?

By Eric Doyle

TRILOGIC'S ROCKET ATTACK

When Trilogic launched its Expert cartridge the boast was that a RAM cartridge can do anything that a ROM could do. Trilogic proved true to its word until the release of Datel's Action Replay Professional IV cartridge with WARP 25 fast disk loader, The Expert just couldn't keep pace.

Now the RAM cartridge is hotting up the cart wars with a new operating system, Expert V3.2R, which includes a new program called Rocket Turboload which intends to take on WARP at its own high speed game.

The principal at stake is that The Expert is based on a RAM chip which can be programmed from disk or tape but Datel's ARP IV is ROM based. This means that an update for The Expert costs the price of a new disk (£3.50) but an update for ARP IV means the price of a new cartridge (currently £34.95).

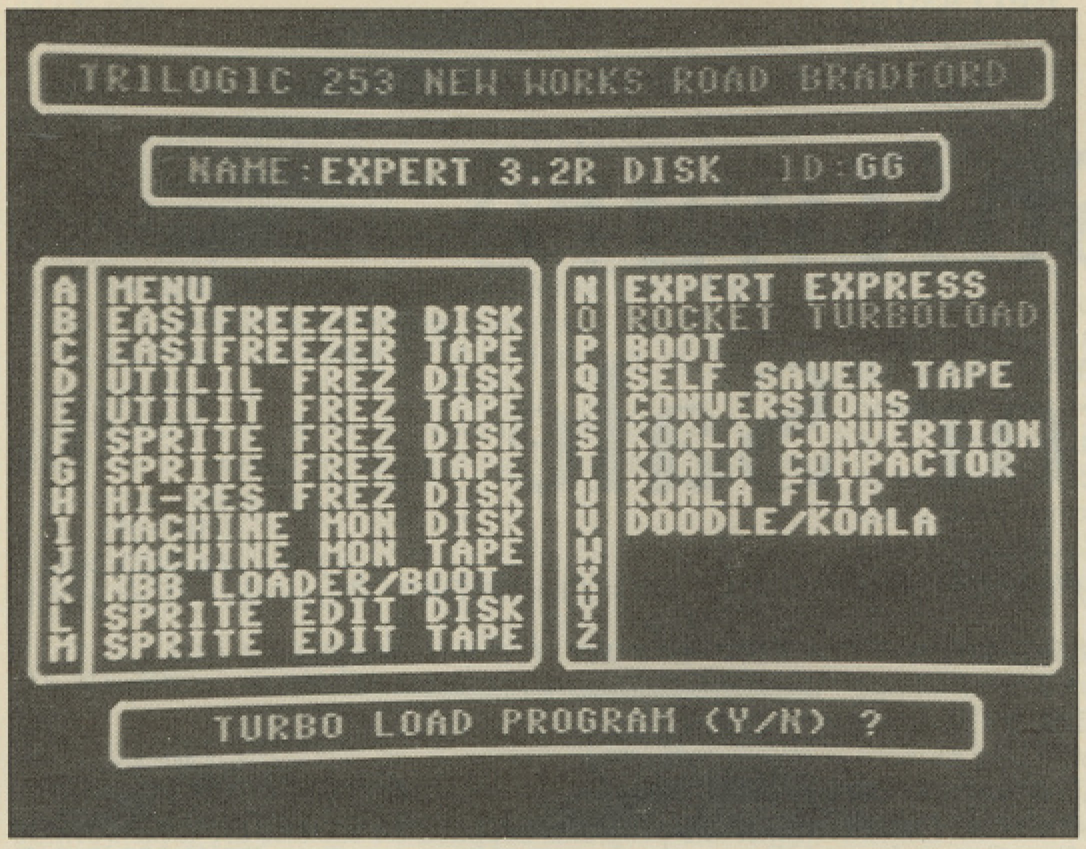
Although The Expert offers many more utilities, the ARP IV cartridge has an extremely fast disk loader which gives it a slight edge for some users.

Rocket Turboload attempts to redress the balance in its favour but how do the two systems compare?

Round One

A program was chosen which filled up 202 blocks of computer memory ($0801-$CFFF). The saving disks were freshly formatted so that the search time for both system's reloaders would be the same. To use WARP all that needs to be done is to plug in the cartridge, load the program for saving, switch the cartridge in and then save. After this a loader program is required which loads as a no block boot.

The Expert is more fiddly to use. First the operating system has to be loaded into the cartridge, then the routine follows a similar path as the WARP system but a final step has to be added. A fresh disk has to be formatted, a second program (Rocket Turboload) has to be loaded, and the program for saving is transferred from



the normal Expert disk to the newly formatted disk through Rocket Turboload.

On balance the first round goes to ARP IV for convenience but Expert scores highly because fewer program conditions trip out the system. ARP IV efficiency against Expert reliability.

Round Two

Compression of the program rendered a reduction from 202 to 172 blocks on ARP IV. Expert initially reduced the program to 158 blocks but after processing through the Rocket converter this increased to 167 blocks. A slight points advantage to Trilogic.

Reloading the program through the ARP IV WARP system always means loading the loader program first. This displays a menu and the cursor has to be moved onto the program required. The RETURN key then initialises the load. The load took about five seconds.

Rocket is easier to use as long as you know the program's name. Using the normal LOAD command followed by "R*progname,8,1 automatically boots the program. This load took about seven seconds.

Although it is obvious that WARP still has a slight edge on speed, the need to select through the loader affects the overall speed depending on how quickly the user loads and runs the menu, selects the desired program and presses RETURN. On balance the second round goes to The Expert for convenience of use.

Round Three

The final round is a measure of the stamina of both the contestants. Which can survive the longest?

Both cartridges offer special 'shaping' utilities through which pokes and infinite lives can be added, and both have respectable memory monitors. Once again The Expert has the edge with joystick port swapping so that all games will use the port of your choice, a Sprite Extractor program to allow the viewing and saving a much improved sprite editor, and an infinite life creator which doesn't work on all programs but will work on many.

The knockout punch is rammed home because The Expert is reprogrammable and there is still room for future additions to the system at low cost. ARP IV additions would have to be added onto a new chip and sold at ten times the cost of an update disk.

Naughty but Nice?

The controversy surrounding cartridges swirls with ever increasing intensity. Are these skeleton keys for use by thieves to unlock protected programs or are they useful teaching utilities which can lay bare the bones of programming structures? The fact that the cartridges can be used to copy most software programs does open up the piracy field but, as Alan Sugar proved recently with his twin cassette decks, it is not illegal to advertise or sell equipment which has a legitimate use.

Let's face it, many programmers and programming houses makes use of The Expert and its kin for developmental and exploratory purposes. Can anyone justify condoning through private use while condemning a product in public? Cartridges MUST never be used for piracy but how can anyone learn progressive programming techniques without a cartridge?

The Expert's Sprite Extractor is a good example of the positive use of cartridges. Using this utility, sprites can be lifted from commercial software for examination of animation routines. The animation characters can then be saved to tape or disk. Using the new Sprite Editor program these characters can be reloaded and altered to create new characters.

The sprite program gives full facilities to view up to eight animation sprites but allows the creation of over 240 in all. The sprites can be animated to check for glitches, viewed in multicolour or hi-res, enlarged, flipped, flopped and generally manipulated into the required shapes. This is a boon to programmers who want to learn animation, few books or magazines can teach the principles in a more graphic form.

Cartridges may be potentially harmful but what is the alternative for serious programmers?

Round Up

For sheer adaptability and flexibility, The Expert is the best possible buy. For convenience it doesn't score so well, the program is stored in a volatile RAM chip and disappears if power is disrupted for too long. The RAM is dynamic and will hold the program if the computer is switched off and on quickly, however.

Every time the cartridge is used the operation system has to be booted up and this is the price that must be paid for cheap upgradibility (well, this and £3.50 for the new disk!)

The addition of the Rocket system has improved The Expert to a point where it can considered to have no equal. Some cartridges may have the edge here and there, but who really cares if one system loads a mere second or two faster than another? What is important is the overall benefits offered and in this department The Expert cannot be beaten. YC

Software for Sale

If you think that one of our programs looks very interesting, but you can't afford the time to type it in then our software service will help you out

It's three o'clock in the morning. You sit at the computer keyboard having just finished a marathon typing session entering one of the superb programs from *Your Commodore*. Your fingers reach for the keyboard and press the letters R, U and N. You press RETURN, sit back and nothing happens.

Everyone has probably faced this problem. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little bug slips through unnoticed.

The *Your Commodore* Software Service makes available all of the programs from each issue on both cassette and disk at a price of £6.00 for disk and £4.00 for cassette. None of the documentation for the programs is supplied with the software since it is all available in the relevant magazine. Should you not have the magazine then back issues are available from the following address:

INFONET LTD, 5 River Park Estate, Berkhamsted, Herts HP4 1HL.
Tel: (04427) 76661

Please contact this address for prices and availability.

The Disk

Programs on the disk will also be supplied as totally working versions, i.e. when possible we will not use Basic Loaders thus making use of the programs much easier. Unfortunately at the moment we cannot duplicate C16 and Plus/4 cassettes. However programs for these machines will be available on the disk.

What programs are available?

At the top of each article you will find a strap containing the article type, C64 Program etc. So that you can see which programs are available on which format, you will also find a couple of symbols after this strap. The symbols have the following meaning:

This symbol means that the program is available on cassette.

These programs are available on disk.

Please Note

Since the programs supplied on cassette are total working versions of the program, we do not put disk-only programs on tape. There is no sense in placing a program that expects to be reading from disk on to tape.

MARCH 1988

LABEL LINKER – Create a library of C128 programs (C128 disk only).
MULTI DUMP – Epson screen dump for C64 owners.
WEOS II – Continuation of program from February 1988. Complete program on this disk or tape.
MUSI LOAD – Play music while your programs load (C64).
PLUS/4 ASSEMBLER – A machine code assembler for Plus/4 users (available on disk only).

ORDER CODE
DISK YDMAR88 £6.00
TAPE YCMAR88 £4.00

APRIL 1988

AUTO START MAKER — Give your disk programs that professional look by making them auto-start (C64 Disk Only).
CLEAR WITH BASIC — A series of short Basic routines illustrating various ways of clearing your C64.
TABULATE — Format your numerical printouts with ease (C64).
Commodore Modem Revealed — A small basic program showing how you can write your own programs for your Commodore modem.
MAKING GEOS BRITISH — Have you ever wished that GEOS had a £ sign or that GEOdex had British addresses — modify your GEOS programs to do just that (C64 disk only).
EXTENDED BACKGROUNDS — Alter the background colour for every screen line — a still display all 256 characters (C64).
GYROSPEED — Transfer your single part programs from C64 tape or disk to turbotape.
ORDER CODE
DISK YDAPR88 £6.00
TAPE YCAPR88 £4.00

MAY 1988

DESIGNER — A powerful drawing package for C64 owners.
80 COLUMN VIDEO — A simple 80 column character designer for the C128 plus disk.
CHARACTER GRABBER — Grab your favourite C64 character sets and use them in your own programs.
ADVENTURE KIT V — The fifth part of our continuing library of C64 adventure writing routines.
C128 FKEYS SWAPPER — Store four sets of function key definitions with this handy utility.
SCREENDRAW 64 — A powerful C64 character and background editor for producing your own scrolling backdrops (disk only).
ORDER CODE
DISK YDMAY88 £6.00
TAPE YCMAY88 £4.00

JUNE '88

DIRECTORY EDITOR — A superb utility that allows you to alter your disk directories as well as add comments (C64 disk only).
30K RAM DISK — Turn unused memory on your C64 into a pseudo disk.
BASIC LISTER — List a Basic program stored on disk without having to load it into memory (disk only).
EASY BASIC TOOLKIT — Give your C64 over 30 new basics – including the ability to store libraries of subroutines.
DISK SECTOR EDITOR — Examine the contents of your Commodore 1541 disks (disk only).

ORDER CODE
DISK YDJUN88 £6.00
TAPE YCJUN88 £4.00

JULY '88

EASIPRINT — Now C16 and Plus/4 owners can access their printer functions with a set of 60 easy to use Basic commands (available on disk only).
HANDY INPUT ROUTINES — Make your Basic programs idiot-proof and more attractive with these two Basic input routines (C64).
MAKING MUSIC — A simple C64 synthesiser to accompany our popular series.
DIY PARALLEL INTERFACE — Use this software and a lead of your own manufacture to drive a Centronics printer (C64).
CP/M CENTRONICS INTERFACE — Give your 128 in CP/M mode access to a Centronics printer..
CASSETTE INLAY PRINTER — Use your printer to produce neat inlays for your cassette collection (C64).
DISKASSEMBLER — Dissassemble your programs direct from disk with this program for the C64 (Disk Only).

ORDER CODE
DISK YDJUL88 £6.00
TAPE YCJUL88 £4.00

Cassettes or disks are available from March 1986. Please ring the editorial office (01-437 0626) for details of these.

ORDER FORM – PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
AUGUST 1988		TAPE (£4.00)	YCAUG88	
AUGUST 1988		DISK (£6.00)	YDAUG88	
OVERSEAS POST £1				
			TOTAL	

NAME...

ADDRESS...

POSTCODE...

I enclose a cheque/postal order for £...... made payable to ARGUS SPECIALIST PUBLICATIONS LTD.

All orders should be sent to: YOUR COMMODORE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS, 9 HALL ROAD, HEMEL HEMPSTEAD, HERTS HP2 7BH.
Please allow 28 days for delivery.

Pagebuilder

A full desktop publisher for the 128 in 80 columns, this unique package allows you to produce documents, forms, letter heads, labels, artwork. It supports the 1750 RAM Packs and 1581 disk drive. Features include selection of fill patterns, line types, border styles, 12 different Fonts, import text and graphics form page illustrator. Ring or write for more details. **Only £49.95**

Page Illustrator

A graphics drawing package, draw using cursor, mouse, lightpen or joystick. Features include, zoom, mirror, fill, border, import text from any word processor, built in label maker, import graphics from PrintShop, PrintMaster and Newsroom graphics. Many more features. Ring or write for details. **Only £34.95**

Disk Drive Problems?

1571/1541 Drive Alignment package by Free Spirit Software reports the alignment condition of the disk drive as you perform adjustments. On screen help is available while the program is running. Includes features for speed and head stop adjustment. Allows you to test each full and half track as you re-align the drive. Complete instruction manual on aligning both 1571 and 1541 drives. Even includes instructions on how to load the alignment program when nothing else will load. Works on the C64, SX64 and 128 in 64 or 128 mode. **Price only 24.95**

SFD1001/8050/8250/4040 Versions available - Ring.

New COMMODORE Version 5 ROM.

Are you experiencing any of the following problems with your 1571 disk drive:-

★ Errors when using Superbase?
★ Device Not Present Errors?
★ Major problems when 2 or more open files?
★ Takes forever to recognize a flippy disk?

If you have one or more of these problems then you need 1571FIX ROM, developed by Commodore Inc USA, do not settle for cheap alternatives, just plugs in, solves over 19 main problems, **only 24.95**

The Next Generation Basic - Basic 8.

We are proud to introduce Basic 8 with Basic 8 Paint, Basic 8 Calc and Basic 8 Write. Basic 8 is the first 128 software package specifically designed to unleash the hidden graphics power of your 128. Using a special wedge technique, Basic 8 achieves performance rivaling that of 16 bit micros! Imagine your 128 in 80 columns producing a resolution of 640 x 200 in mono and 640 x 192 in 16 colours without any additional hardware!

Basic 8 adds over 50 Hi-Res graphic commands to Basic 7 and is completely compatible with it's advanced non-graphic commands. You work in true 3-Dimensional environment, controlling such parameters as perspective viewing angles and the origin & depth of the view, and many many more features, **only 34.95**

Basic Paint, Calc and Write were all written under the new Basic 8 environment. These provide working examples of the power of this unique package. Basic 8 is available on a chip and a 64k Video RAM upgrade is available.

READ AND WRITE IBM FILES ON YOUR 128 OR 64.

The Big Blue Reader is ideal for those who use IBM PC compatible computers at work and have a 128 or 64 at home. The Big Blue Reader is not an emulator, but rather it is a unique and easy to use file copy program designed to transfer word processing, text and ASCII files between two totally different disk formats, the Commodore 128/64 and MS-DOS disk formats. The Big Blue Reader requires a 1571 on a 64 or 128 and cannot use a 1541. The Big Blue Reader **only 29.95**

Timeworks Software.

The popular collection of business for the 128 is now available at the new retail price of **39.95 each.** WordWriter 128, a full featured 40 or 80 column word processor. SwiftCalc 128, spread sheet with Sideways printing. DataManager a complete database.

Financial Systems Software Ltd

2nd Floor, Anbrian House, St Mary's Street,
Worcester, WR1 1HA.
Telephone (0386) 750217
All prices include VAT and P&P
Foreign Orders add £2.50 per package.

VISA

---

commodore

AMIGA — FROM £425!

Amiga A500 with 512K RAM, 880K 3.5" disk, mouse, software	£425
Amiga A500 TV As above, plus A521 TV/composite modulator	£445
Amiga A500M with A1081 high-res colour monitor	£675
Amiga B2000 with 1MB RAM, 880K 3.5" disk, mouse, software	£999
Amiga B2000M As above, plus A1081 hi-res colour monitor	£1245
Amiga B2000 XTM As above, plus PC XT bridge board & floppy	£1745

Add £50 for A1084 in place of A1081. Prices include 15% VAT, a FREE copy of DeLuxe Paint, FREE delivery to your door, one year warranty. Overnight delivery £12 On-site maintenance option available.

PERIPHERALS AMIGA

A2088 PC-XT bridge board with 5.25" internal disk drive	£499
A2286 PC-AT bridge board with 5.25" internal disk drive	£699
A2092PC 20MB internal hard disk with MS DOS controller	£525
A2092A 20MB internal hard disk with Amiga DOS controller	£625
A521 TV/composite modulator	£25
512K plug-in RAM/clock (A500)	£99
A1081 high-res colour monitor	£275
A1084 high-res colour monitor	£325
A1010 3.5" external disk drive	£185
Cumana 3.5" external disk drive	£129
NEC internal drive (2000 only)	£129
A2010 internal drive (2000 only)	£175
Genlock B2000 video controller	£225
A2052 2MB internal RAM card	£375
A2094A 40MB Amiga Dos hard disk	Call
Supra hard disks 20MB - 250MB	Call
Citizen 120D 120 cps, F/T, NLQ	£185
Star LC10 120 cps, F/T, NLQ	£225
Precision printer 480 cps, NLQ	£399
Canon PJ1080A colour, 80 dpi	£499
Xerox 4020 colour, 240 dpi	£1125
HP LaserJet laser, 300 dpi	£1995

CREATIVITY AMIGA

De Luxe Paint II (PAL)		£59.95
DigiPaint 4096 colours		39.95
De Luxe Music (PAL)		59.95
Page Setter		94.95
City Desk		89.95
Animator Apprentice		179.95
ProWrite	Wordprocessing fully integrated with graphics	89.95
VizaWrite Desktop	High performance desktop publishing wordprocessor	79.95
Word Perfect	Amiga version of the No.1 best selling wordprocessor	199.95
The Works	Analyze + Organize + Scribble, integrated desktop	139.95
Publisher Plus	Enhanced version of the original desktop publisher	99.95
Flow	The Idea Processor from New Horizons Software	69.95
PixMate	Professional image enhancement straight from NASA!	44.95
Prism Plus	4096 HAM colour paint pack 1024 x 1024 (needs 1MB)	54.95
Calligrapher	Professional font editor, fonts up to 160 x 256 pixels	74.95
De Luxe Video II	Create all types of video presentations with ease	59.95
Aegis VideoScape	Full 3D animations in all resolutions with overscan	124.95
Pro-Video CGI	Professional video titler with fonts, extra fonts available	159.95
Aegis Animator	Full storyboarding package, includes Aegis Images	94.95
TV Show	PAL image presentation in IFF or HAM with overscan	89.95
DigiPic	Fast frame-grabber, 32 colours or grey scale	299.95
Art of Chess	The best Amiga chess yet, packed with features	24.95

PRODUCTIVITY AMIGA

SAM Basic		£89.95
A/C Basic		154.95
A/C Fortran		229.95
Lattice C 4.0		144.95
Lattice C Professional		299.95
MCC Pascal		69.95
64 Emulator	Use your C64 programs and files! Includes cable	69.95
SuperBase Personal	Relational database power, without programming!	79.95
SuperBase Professional	With Forms Editor and DML program language	194.95
Acquisition (V1.3)	Relational database with programming language	224.95
Maxiplan 500	Ultimate Amiga spreadsheet, + text/graphics/speech	89.95
Maxiplan Plus	As Maxiplan 500 above, plus powerful Macro language	129.95
VIP Professional	1-2-3 compatible spreadsheet + database & graphs	99.95
Logistix	2048 x 1024 spreadsheet, database, graphs & calendar	99.95
Businessware Pack 1	Invoicing, Sales Ledger, Stock Control, by Panmead	159.95
Businessware Pack 2	General Ledger, Purchase Ledger, by Panmead	159.95
Intro Cad	Entry level CAD package, printer or plotter output	54.95
Aegis Draw Plus	Comprehensive CAD with multiple layers & windows	124.95
X-Cad	Professional CAD system (needs 2MB)	424.95
MCC Assembler	Professional development system with linker & libraries	59.95

128

Commodore 128D		£399
Commodore 128		199
NEW 1581 disk drive 3.5"		199
Commodore 1571 disk drive		199
Commodore 1541C disk drive		169
NEW Commodore PC-III		Call
VizaWrite 128 Classic		59.95
VizaWrite 64 (disk)		39.95
VizaWrite 64 (cartridge)		49.95
VizaStar 128		79.95
VizaStar 64 XL8		59.95
VizaStar 64 XL4		49.95
SuperBase 128		£29.95
SuperBase 64 & Plus/4		24.95
SuperBase: The Book		11.95
SuperBase Starter 64		14.95
SuperScript 128		29.95
SuperScript 64		24.95
PetSpeed	Oxford Systems' best selling Basic compiler (state 64/128)	19.95
Oxford Pascal	Complete J & W Pascal, with extensions (state 64/128)	19.95
Oxford Pascal 64 (tape)	As above, for C64 tape users	19.95
Cobol	Complete COBOL package from Abacus (state 64/128)	29.95
Super C	Complete C language to K & R standard (state 64/128)	39.95
Hack Pack	All-in-one programmer's Toolkit/Ram-disk (state 64/128)	19.95
Oxford Basic 64	All-in-one Basic compiler, Toolkit, compressor	19.95
1750 Expansion RAM	512K Expansion RAM cartridge for the C128	149.95
RamDos 128	Lightning-fast RAM-disk for 512K expansion RAM	29.95
1764 Expansion RAM	256K Expansion RAM cartridge for the C64	99.95
Anatomy of the 128	The 500-page insider's guide to the 128	9.95

IF YOU WANT IT TOMORROW... CALL US TODAY! ON 0483 - 504125

Prices are POST FREE & include VAT. Order by phone with your credit card, or send cheque/PO or your credit card number. Official orders welcome. We despatch same day by FIRST CLASS post. Please allow 5 days for delivery of hardware orders. Prices quoted are subject to availability. Ref. A58

VISA

Calco Software

LAKESIDE HOUSE, KINGSTON HILL, SURREY, KT2 7QT. TEL 01-546-7256

JINKS

Imagine a cross between Breakout and Pinball, set over a horizontal scrolling playing area and you will have some idea of what Jinks is all about. Needless to say though, the software industry prefers dressing mutton up as lamb and there is the usual inane scenario.

You have to guide your scout craft across the surface of the planet Atavi in what is described as a unique exploratory research mission. The planet is beautiful yet remains an enigma. I'm not surprised seeing that it still resembles a cross between a pinball machine and a breakout game.

Jinks has been released simultaneously on both the C64 and the Amiga and so both formats can usefully be compared.

The object is to guide your probe around the planet using your spaceglider. Translated, this means hit the ball with your bat. Points are awarded for hitting bricks; sorry objects that happen to be lying around. As well, there are special objects such as mini worlds that when hit, can help your cause by adding extra lives, restoring your bat size, etc.



The bat is straight on one side and angled on the other. Flipping between these two options allows you to control the path of the ball. In practice, the ball tends to disappear off screen and you spend large chunks of the game chasing after it. This is by far the worst implemented part of the game. One of the hazards that needs to be avoided is letting your ball fall through a hole in the ground. Frequently, I found that I was getting 'game over' messages with the ball nowhere in sight because I couldn't move the bat quickly enough to catch up. This is especially true on the C64 version which is rendered largely unplayable.

As you move along the playing area, you have to avoid touching any of the moving objects. The first touch reduces your bat to half its usual size. The second touch kills you. Bat size can be restored by guiding the ball onto the appropriate object.



There are four different screens to try your luck on. Guiding your ball onto a rotating ball, special objects, take you onto a 'bonus screen' although no bonus is given to your score. From here, you can select any of the other screens to visit by guiding your ball into the appropriate slot. Return visits to previously encountered screens results in more moving objects to avoid and more holes to fall through. There are effectively nine different skill levels as you can alter the speed of the ball and the effects of gravity upon it.

Graphics look quite slick on the Amiga although each screen takes an age to load in. Conversely, on the C64, all the screens are in memory at the same time but they look fairly rough.

Despite what I have said, the idea behind Jinks is quite a good one and given more thought in the gameplay, it could have been a good game. As it stands, the Amiga version is average while the C64 version is quite dreadful.

G.R.H

Touchline:

Title: *Jinks.* ***Supplier:*** *GO! Unit 2/3 Holford Way, Holford, Birmingham B6 7AX.* ***Price:*** *CBM 64 – £9.99 (Ca) £11.99 (Disk) Amiga – £24.99.*

	C64	Amiga
Originality:	7/10	7/10
Graphics:	3/10	7/10
Playability:	3/10	5/10
Value:	3/10	6/10

JINKS

Relative File Programming

Sort out your troublesome relatives

By Eric Ramsay

Following the first three articles in this series we are now at the stage in which a relative file has been set up and records entered into it. These can be displayed in a browse mode, amended and even printed out. At the moment, however, they are sorted in any order, appearing as they've been written into the file.

The first record you entered will be record number 1, the last entered record is the last to appear on the screen. Obviously, a file such as this has only limited use, so we must decide on a keyfield routine, and then a sort which will show the records in order according to the keyfield.

The first routine we need is one which will declare the keyfield for the file. You have the option of making this permanent, but I have always programmed so that the keyfield may be changed so that the records can be sorted in any field. First of all the user has to be able to change the keyfield, and to do that it would be a good idea if we displayed the field details for him to look at while he decides.

Declare keyfield routine:

```
400 print"clrscreen": rem **** clear the screen
410 print"FIELD"; tab(?); "Name"; tab(?)"Length"
420 print:for a=0 to c
430 printa+1;tab(?);FIELD$ (a); tab(?); FIELD (a)
440 next a: print:print"Which FIELD to key?"
450 gosub3430: rem *** input number Routine
460 kf=n-1:
470 gosub 4370-rem *** branch to sort Routine
480 goto (? Wherever you have the file menu)
```

So far we have the number of the desired field to be keyed in the variable KF. One advantage of doing it this way is that just after a file has been set up and before this routine has been run for the first time, KF will be 0, which happens to be the number of the first field.

Among all the other statements at the beginning of the program, you might like to create a graphic image of the KEY, with two left-graphic R and a 0, like this:

```
10kf/[cR][cR]
```

I chose the 0 for the graphic instead of the right-graphic on the W key because this graphic symbol does not work in lower case; you get a W instead of the round symbol. You might like to show the graphic symbol in the request for the keyfield number:

```
440 next a:print:print "Which FIELD to ";kf$
```

Now we have set up the keyfield variable, it is used inside the field array of the current file, then by storing the keyfield string from each record into an array to be sorted. We will call the sorting string array RSRT$(). Since this will be an array, it will have to be DIMensioned in the beginning of the program much like all the others we have already dealt with.

Dynamic Arrays

A word of warning – you cannot DIMension this or the other sort array which we will come to in a moment, using a real number. If we did so, you would only have to write a new record over that number and you would get a syntax error. For this reason, you must use a dynamic array because you cannot tell for each program file how many records you might be going to use, hence the variable number for the DIM. Which variable you use in the dynamic array depends on you.

If you program to halt the run at the 49th new record created, resave the housekeeping, CLR and then re-run the program; you would be able to use the RN variable inside the DIM:

```
dim rsrt$(rn+50)
```

Or, simpler but setting aside much unused variable space, is to use the very maximum number of records that can be written, the AVAIL variables:

```
dim rsrt$(avail)
```

after which every possible record you might enter has already been

DIMensioned for. You would never get the dreaded 'Bad Subscript Error' message with that, but you might have 100 records in the file and have the arrays DIMensioned for 4000!

The method I use is to POKE all the variables to a free page of RAM, starting at DEC 4864. Everything, field names, lengths, RN and all the other variables not yet mentioned are POKEd there. When RN has increased by 49 from when the program was last RUN, the program executes a CLR and then peeks all the location in memory to get the variables back. The arrays are then re-DIMensioned. The housekeeping is then saved as a BSAVE to disk for the next time the file is accessed in a program RUN.

With this method it is not possible to exceed the DIMensioned numbers in the arrays, but neither is it necessary to set aside huge amounts of variable memory when it is not required. However, I digress slightly.

You must have DIM'd the array RSRT$ and RSRT%, either with AVAIL or RN+50 or 100. Now we will write the first of the sort routine. We have to read the keyfields of all the records currently in the file, (and deleted ones) into the string array RSRT$, then sort them into alphanumeric order, ending up with the actual record numbers stored in the array RSRT%.

Why the %? Well, in case you have not used integer arrays before, they save much space on the disk when they are stored. An ordinary number is stored to several decimal places, which is wasteful of space, but an integer is all we need for this application (since there is never a need to read record number 4.667)!

Let us begin the routine. You must have opened the file. The screen is cleared, and the message placed to inform the user what is happening. I have also programmed a display of the record number which is being processed so that the user has something to watch. Time seems to pass more quickly while you watch a number change!

```
4360 rem **** Sort RECORDs routine ******
4370 print(clearscreen): print "RECORD Processing Now"
```

The loop, 1 to the current record number used plus, most important, the variable storing the number of records currently deleted is added to it, DREC and then the cursor is HOMEd (without clear) to show the record being processed, which of course, since PO is the loop name, is PO.

```
4380 forpo=1 to rn+drec:print (home): printpo
```

Then the record to be read variable is declared as PO: BASIC 2, you have your own, don't forget:

```
4390 rrec=po:drec=0: gosub 4130:rem ****drec is delete rn
4390 (BASIC 2 users) rp=po: gosub 4114: gosub 4130
```

Although we haven't seen this yet, the following line will be explained in greater detail later. Basically, the loop will detect records deleted, and will find the set character, CHR$ (255). Since this will mean nothing to the sort routine, we declare the sort string for this record as the lowest possible value, a String of 'Z'...

```
4400 if disk$= chr$(255) then drec=drec+1: rout% (drec)=po: rsrt$(po)= "ZZZZZZZZZZZZ": goto 4420
```

ROUT%() is the array which stores the deleted record numbers for future use. They are stored at the time of deletion, but if this is a new sort, the sorted record numbers change, and you would end up overwriting good records if this array was not updated! Now we can look inside the DISK$. Notice there is no gosub to the field string routine. What for? That would merely slow things down. Instead, MID$ and the field pointers array with the length of the field array, FIELD() is used to pick out the field to sort, KF. Then, because this field might be 50 characters long, LEFT$ is used to take the left 12 characters to sort. The loop is then ended. You may use more characters if you wish by increasing this number inside the LEFT$.

```
4410 rsrt$(po)=left$ (mid$(disk$, fpoi (kf),
FIELD (kf)),12)
4420 nextpo
```

At the end of this loop, the entire field has been scanned, the relevent field has been stored in the array RSRT$ and we are ready to Sort.

FIELDS SEPARATED SCAN FOR SORT – ALL BASICs

Yes, don't panic. You should have realised that this routine would not work for you. You require a separate read routine. You will remember that your read record routine reads the record inside a loop, and ends with the record neatly inside the array DISP$(). This routine could be used, except that it is a waste of time reading the entire record when you only need to look at the single field. So if you have used the field separated method, you may take advantage of the fact that you can only read one field at a time. For this application this method has an advantage. Make these changes to the above:

```
4390 drec=0:rem **** DO NOT use the Read Routine. 4130
4390 (BASIC 2 users) rp=po:gosub 4114
4392 RECORD #5, (po), (fpoi(kf): gosub 3610
4394 RECORD 5, (po), (fpoi(kf): gosub 3610

BASIC2 users:

4392 print #5, "p"+chr$ (101)+chr $(lo)+chr$(hi)
+chr$ (fpoi (kf):gosub 3610
4394 print #15, "p"+chr$ (101)+chr $(lo)+chr$(hi) +chr$ (fpoi(kf): gosub3610
```

Then read the keyfield into the sort string array RSRT$(), no need to use DISP$.

```
4396 input #5,rsrt$(po):gosub 3610
```

And you have achieved the same result: the keyfield of every record in the file is stored in the string array RSRT$. The RSRT$ for records which have been deleted are stored as a string of Z characters, making sure that they will be right at the end of the sorted array.

You may like at this stage to store the keyfield string array into another sequential file, to which new records as they are entered into the file may be added. This saves scanning the file every time the file is sorted, but you might not wish to do this. Close the relative file, and use this routine to store the sorted string array:

```
4430 n3$=left$ (name$+. keydata+...., 16): dopen #3, "@" + (n3$), d0, u8, w: gosub3610
4400 forpo=1 torn+drec: print #3, rsrt$(po): nextpo
```

```
4450 Dclose
```

Now, because this routine will be used for sorting records without scanning the file, the next line skips the INPUT of the sort string array. I shall come to that later.

```
4460 goto4510:rem **** skip read file
```

Now we go into the sort routine. This is the only routine which is not my own. It is based on a routine called 'PetSort' which is public domain and written by Nick Marcopoulos. In fact the original needed some working on, since it suggested that the record number to the sort string and then stripping off the record numbers after the sort was finished, then reading the sorted array into another array for the sort result.

This was all quite cumbersome and actually unnecessary, since the effect of the sort is to shift the pointers in the array RSRT% and does not move the actual data contained in the strings at all.

I removed the need of any second array, which when you might have 4000 records is a terrible waste of variable space, and the routine works very well for this application. Double check that you have typed it in correctly or the routine will simply not work at all.

```
4510 print(clrscreen) "Sorting now... Please Wait."
4520 n=rn+ drec:rem *** DREC will be explained later
4530 fori=1ton:rsrt%(i)=i:next
4540 w=n/2.6:k=5: ifn=0 then return
4550 ifw > thenk=17:ifw > 17 thenk =59: ifw>59 then k=199: ifw>199 thenk=673
4560 k=int ((k+1/3.38): form=0tok-1
4570 fori=m+k to n step k:w=rsrt%(i): for j=i-k to m step-k
4580 if rsrt$(rsrt%(j)) > rsrt$(w) then rsrt% (j+k)=rsrt%(j) (j):next
4590 rsrt%(j+k)=w:nexti:next:ifk > 1 then 4560
4600 return
```

Now the records are sorted, but the effect will not be apparent until you have changed the lines which request the records read routine. At present they are simply reading the record number of the file, not the sorted record number. In case you find this difficult to grasp, let me explain this further. Your records are presently on the disk, filed in the order in which you typed them in. They look like this:

```
RECORD

1(    ) 2(    ) 3(    ) 4(    )
5(    ) and so on. But the data they contain might be:

1(  5) 2(  2) 3(  1) 4(  3) 5(  4) and so on.
```

At present when you request record number three in the file, you always see the actual record three. Now we have sorted them we have the sorted order in an array: the array for the above RECORDs would be:

```
RSRT% (RECORD REQUESTED)=ACTUAL RECORD
RSRT% (1)=3
RSRT% (2)=2
RSRT% (3)=4
RSRT% (4)=5
RSRT% (5)=1
```

You must remember that any array is really a label: when you request a number inside the brackets of an array you are asking for the label of a particular package, which is something completely different. A browse of the file shown above, using the sorted record array RSRT%() would look at record number three first, then record two next, then record four, and so on.

You will be unaware of this because you will see record number 1 on the screen while the file record, that actually read, will be three. If you want to see the file record number you are actually accessing, you should add this to the screen display routine:

```
4300 print"Sorted RECORD No: ";disp;" of":
;rn;"Virtual"; rsrt%(disp)
```

In the new part of the line, we have used the sorted array RSRT% in reverse to get the virtual, the file, record number.

Now the effect on the file has been to arrange the records for display in alphanumerical order. But why did we want the deleted records to store a "ZZZZZZZZZZZZ" as the RSRT$? Simple. The loop included not only current record numbers, but also deleted records. We shall go into more detail in the next article but if you have declared a loop, FOR PO=1 TO 5+2, then PO Loops 7 times. But if you read back a string created inside that loop using just 5; FOR PO=1 TO 5 then the loop will only read the first 5. Now the string containing "ZZZZZZZZ" will have been sorted to the end of the array and will not be displayed because they are outside the RN range.

Now you must go through your routines carefully, and change every request for a File Read or Write.

```
WREC=RN or WREC=DISP
to
WREC=RSRT% (RN) or
WREC=RSRT% (DISP)
and of course, for the BASIC 2 users,
to
RP=RSRT%(RN) or
RP=RSRT%(DISP)
```

before gosub to the record pointers routine. To help you, here is a list of the relevant line numbers. Basic 2 users, you declare RP as the relevant RSRT%() before you branch to the record pointers routine.

In the enter records routine, the write record reads the record number as RN, which is correct. But because this record is a new one the sorted array, RSRT%() does not exist for this record. Since the enter records routine adds new records to the end of the file until the next sort, the new record sorted number will be the end record number, which here is RN. Thus we add:

```
RSRT%(RN)=RN

to the Line 2580:

2580 wrec=rn:rsrt%(rn)=rn
```

Now we add the RSRT% to the browse record routine so that the record accessed is the sorted record.

```
2740 rrec=rsrt%(disp): rem **** get RECORD number

We amend the edit record routine, otherwise we will be amending the wrong record!

3000 wrec=rsrt%((disp)): gosub 4070

Then the lines in the copy routine:

3140 rn=rn+1: wrec=rn: rsrt% (rn)=rn
3170 disp=rn:rrec=rsrt% (rn): fin=fin+1
```

And if you wish, you may show the virtual record number in the hardcopy routine from the browse menu:

```
3210 print # 4, "RECORD Number
";disp;" of ";rn;"
Virtual: "; rsrt%(disp)
```

Finally, now that you have a keyfield you might like to display on the screen and on the hardcopy of the file which field is the key. The following lines and changes will do that for you.

```
3230 ifkf=a thenprint # 4,kf$;
:elseprint # 4, ;
4310 print:print:for a=0toc: ifa=kf then
printtab (3); kf$; :elseprint" ";
```

Now we have to amend the housekeeping file. We need to add a short routine to store and read back the sorted record numbers, as well as the new program control variables. There are more of those to come yet. Here are the complete routines. BASIC 2, I am sure it is not necessary for me to write them for you as well. Just add the new line numbers, the open statements are the same.

```
200 n1$=LEFT$ ("HSE.
"+NAME$+"........", 16)
210 dopen@3, " # "+(n1$), w:gosub
3610
220 print # 3,c,rn,kf: gosub 3610
230 for a=0 to c
240 print # 3,FIELD$(a): gosub 3610
250 print # 3,FIELD(a) gosub 3610
260 nexta
270 forpo=1torn
280 print # 3,rsrt% (po)
290 next po
300 dclose:gosub 3610
340 return
And the Read Housekeeping.

350 n1$=LEFT$ ("HSE"+NAME$+"
........", 16)
360 dopen # 3, (n1$), r:gosub 3610
370 input # 3,c,rn,kf: gosub 3610
380 for a=0 to c
390 input # 3,FIELD$(a): gosub 3610
400 input # 3, FIELD(a): gosub 3610
410 nexta
420 forpo=1torn
430 input # 3,rsrt%(po)
440 next po
450 dclose
480 return
```

Searching on the Keyfield

Now you have gone to all this trouble to sort the records, you may take advantage of it. After the sort, the records appear on the screen, in alphanumeric order as you browse, one after another through the file. This is all very well, but if you have 4000 records, it would take a lot of browsing to find the particular entry you need.

Of course, you could use the goto option on the browse menu, but locating the individual record would still take some time. What we need is a search on the keyfield. This is the search routine I promised in a previous article, which will find any entry in the keyfield in 9 or less reads from 1000 records!

How does it work? It is actually quite simple. Imagine you had a card file of names in alphabetical order. There is no marking in the card file showing where each letter begins or ends; how would you look for one name in that file?

You would look at a card about the middle of the cardfile, and compare the name there with the name you are looking for. If the name is 'less than' or a lower alphabetical name, you would look in the previous half of the file for another sample. If the card you selected was higher, you would look in the next half of the file. When you get close to it, you would look at individual cards until you found the name you were looking for or you knew it was not present.

Keyfield Search

Do not forget, you must have opened the relative file for accessing. BASIC 2 users, you must have opened the Error Channel 15.

First, the once extra read counter, the NUdge counter is set to 0. This will be explained in a moment. The request string is inputed into n$.

```
3030 nu=0:yy=0
3040 print"Enter Search Data:"
3050 input n$
```

Now we come to the processing. BN is the beginning number which is the nuber of records divided by 2, in other words, the routine find the middle of the file. The BIN variable is the number which will be processed from now on.

```
3060 bn=rn/2:bin=bn
```

The record number is assigned into RREC. Note the RSRT%() array find the sorted record number. This is important.

```
3070 rrec=rsrt%(int(bin))
```

```
BASIC 2:

3070 rp=rsrt%(int(bin)):gosub 4114
```

Then the record is read. Field separated method users would probably be as well to copy the single field read method I used in the third article for the sorted keyfield string RSRT$: or to write their own subroutine.

```
3075 RECORD # 5, (rrec), (fpoi(kf)):
gosub 3610
3077 RECORD # 5, (rrec), (fpoi(kf)):
gosub 3610
3080 input # 5, (disp$(kf)): gosub 3610
```

Using the instr function the keyfield string is searched for the request string. If the match is found then the RREC is declared as the number for reading and the display number is quoted for the display record routine. (Otherwise there will be blanks on the RECORD screen.) The counter to show a positive search result, YY, is set to 1, and then the routine branches back to the display part of the browse routine so that the matched record is displayed.

```
3080 gosub 4120: gosub 4180:ifinstr
(disp$(kf),n$)
>0then rrec=rsrt% (int(bin)):disp=int
(bin): yy=1: goto 2750
```

But the record read might not provide a match. In that case the routine continues into 3090, where the BN variable is again halved, having the effect of quartering the file. A quick check is made in case the BN variable has been halved to less than 1. If it has then the nudge variable is set to one for one last try. This is because some files will have many records with very similar keyfields, taking a disproportionate number of entries in a particular part of the file. The nudge allows one more read.

```
3090 bn=bn/2:ifint(bn)<1 then
nu=nu+1:bn =1
```

If the nudge variable has already been set to one, then the routine ends, displays the 'file not found' message, and returns to the browse with the record number changed to the nearest record found to the match request. This is useful to move the file display to a particular part of the file even when you know the requested string will not be found.

```
3100 if os>1 then print "File Not
Found:" :sleep 2
:rrec=rsrt% (int(bin)): disp=int (bin):
goto 2750
```

Now we do the comparisons: if the requested search string, N$ is 'greater than' i.e. a higher alphanumeric value, then the keyfield string just read, then the halved processing number BN is added to the BIN number to give the next record number to inspect, and the routine branches back to the next record read.

Likewise, if the value of the string read is 'less than' the string requested then the halved number BN is deducted from the BIN number. In this way the file is repeatedly halved until the record is found or the nudged variable is greater than 1, showing that the record is not present in the file.

```
3110 ifn$ > disp$(kf) then
bin=bin+bn:goto 3070
3120 ifn$ < disp$(kf) then bin=bin-
bn: goto 3070
```

You remember that a display counter YY was set to 1. This is indented to display the keyfield of the searched record in reverse, (or underline, Commodore 128 80 column users). If you wish, you make like to change the lines in the display records routine to effect this.

```
4320 ifyy=1anda=kfthenprint; "(in here
place reverse or underline character)";
4350 nexta:yy=0:return
```

Basic 2 Keyfield Search

Without the very fast and useful INSTR routine, BASIC 2 users will have to use their own small comparison routine at the end of the serach. The lines are as follows:

```
3082 gosub 3122: if sr=1 then
rrec=rsrt% (int(bin)):
disp=int(bin): yy=1: goto 2750
```

The remainder is the same except that you need this small subroutine to compare the strings, and to return the result SR.

The search result is first set to 0. If the length of the search string is longer than the length of the field string to be searched it would be a waste of time to continue, so if this is the case then the routine returns:

```
3122 sr=0: iflen (n$)>len (disp$ (kf))
then return
```

Then the search loop is started to the length of the field string minus the length of the search string:

```
3124 fors=1tolen (disp$(kf))-len(n$)-1
```

Using MID$, the field string is scanned from left to right until there is one character less than the length of N$. If a match is found then the loop is closed and the SR result is set to 1:

```
3126 ifmid$ (disp$(kf), s, len (n$)=
thens=len
(disp$(kf))-len (n$)-1: sr=1: return
```

After the loop has finished, DISP$ has been searched without result, so the routine RETURNs with SR as 0:

```
3138 nexts:return
```

There is the keyfield search. In the next article I will provide the routines for deleting records, and a sequential search method which will find any data anywhere within a file. YC

Listings

Get it right first time with our deluxe program system for the C64.

You may have noticed that our listings are free of those horrible little black blobs which send you searching around the keyboard for a suitable graphic symbol. You may also have noticed the funny numbers by the side of each line of the listing. Fret no more, it's all part of our easy entry aid.

Instead of those nasty graphics and rows of countless spaces in PRINT statements and strings we use a special coding system. The code, or mnemonic, is always contained in square brackets and you'll soon learn to decipher their meanings.

For example, [SA] would mean type in a Shifted A, or an ace of spades in layman's terms, and [SA10] would mean a row of ten of these symbols.

[S+2] means hold down the shift key and press the plus key twice. It doesn't take a great leap of logic to realise that [C+2] means exactly the same thing except that the Commodore key (bottom left of the keyboard) is held down instead of the shift key.

If more than two spaces appear in a statement then this will be printed as [SPC4] or, exceptionally, [SSPC4]. Translated into English this means press the spacebar four times or in the latter case hold the shift key down while you do it.

A string of special characters could appear as:

[CTRL N, DOWN2,LEFT5,BLUE, F3,C3]

This would be achieved by holding down the CTRL key as you press N, press the cursor key down twice, the cursor left key five times, press the key marked BLUE while holding down the CTRL key, press the F3 key and, finally hold the Commodore key down while pressing the number two key (C2 would of course make the computer print in brown).

Always remember that you should only have a row of graphics characters on your screen with no square brackets and no commas, unless something like this appears:

[SS],[C*]

In this case the two characters should have a comma between them.

On rare occasions [REV T] will appear in a listing. This is a delete symbol and is created by entering the line up to this mnemonic. Then type a closing quotation mark (SHIFT & 2) and delete it. This gets the computer out of quotes mode. Hold down CTRL and press the number nine key (RVSON), type the relevant number of reversed T's and then hold down CTRL and press zero (RVSOFF). Next type another quotation mark and delete it again. Now finish the line and press RETURN.

A list of these special cases is given in the table but remember that only one of these mnemonics will appear outside of a PRINT string: the symbol for pi. This may appear when its value is needed in a calculation so this may look something like:

:CC=2*[PI]*R:

Ignore the square brackets and just type in a shifted upward pointing arrow (ie. the pi symbol).

```
PROGRAM: SYNTAX CHECKER

5 REM SYTAX CHECKER - ERIC DOYLE

10 BL=10    :LN=70    :SA=49152
20 FOR L=0 TO BL:CX=0:FOR D=0 TO
 15
30 READ A:IF A>255THENPRINT"NUMB
ER TO LARGE";LN+(L*10):STOP
40 CX=CX+A:POKE SA+L*16+D,A:NEXT
 D
50 READ A:IF A><CX THENPRINT"ERR
OR IN LINE";LN+(L*10):STOP
60 NEXT L:SYS 49152:NEW
70 DATA 173,5,3,201,165,208,31,1
20,169,9,141,32,208,141,33,208,1
847
80 DATA 169,7,141,134,2,169,13,3
2,210,255,169,64,141,4,3,169,168
2
90 DATA 192,141,5,3,88,96,120,16
9,124,141,4,3,169,165,141,5,1566

100 DATA 3,169,14,141,134,2,141,
32,208,169,6,141,33,208,88,96,15
85
110 DATA 32,124,165,72,138,72,15
2,72,162,0,165,20,133,254,165,21
,1747
120 DATA 24,101,254,133,254,189,
0,2,240,18,69,254,133,254,232,18
9,2346
130 DATA 0,2,240,8,24,101,254,13
3,254,232,208,233,169,1,141,134,
2134
140 DATA 2,165,254,74,74,74,74,3
2,156,192,32,210,255,165,254,41,
2054
150 DATA 15,32,156,192,32,210,25
5,169,13,32,210,255,169,13,32,21
0,1995
160 DATA 255,169,7,141,134,2,104
,168,104,170,104,96,24,105,48,20
1,1832
170 DATA 58,16,1,96,24,105,7,96,
0,0,0,0,0,0,0,0,403
```

by Eric Doyle

Checksum Program

The hexadecimal numbers appearing in a column to the left of the listing should not be typed in with the program. These are merely checksum values and are there to help you get each line right. Don't worry if you don't understand the hexadecimal system, as long as you can compare two characters on the screen with the corresponding two characters in the magazine you can use our line checking program.

Type in the Checksum Program, make sure that you've not made any mistakes and save it to tape or disk immediately because it will be used with most of the present and future listings appearing in Your Commodore.

At the start of each programming session, load Checksum and run it. The screen will turn brown with yellow characters and each time you type in a line and press the RETURN key a number will appear on the screen in white. This should be the same as the corresponding value in the magazine.

If the two values don't relate to one another, you have not copied the line exactly as printed so go back and check each character carefully. When you find the error simply correct it and press RETURN again.

If you want to turn off the checker simply type SYS49152 and the screen will return to the familiar blue colours. You can then do whatever it was you wanted to do and if this doesn't use the area where Checksum lies you can go back to it with the same SYS command.

No system is foolproof but the chances of two errors cancelling one Many of the listings are presented in lower case. To turn your computer to lower case mode press the Commodore key and the SHIFT key at the same time.

YC

Mnemonic	Symbol	Keypress
[RIGHT]		CRSR left/right
[LEFT]		SHIFT & CRSR left/right
[DOWN]		CRSR up/down
[UP]		SHIFT & CRSR up/down
[F1]		f1 key
[F2]		SHIFT & f1 key
[F3]		f3 key
[F4]		SHIFT & f3 key
[F5]		f5 key
[F6]		SHIFT & f5 key
[F7]		f7 key
[F8]		SHIFT & f7 key
[HOME]		CLR/HOME
[CLR]		SHIFT & CLR/HOME
[RVSON]		CTRL & 9
[RVSOFF]		CTRL & 0

Mnemonic	Symbol	Keypress
[BLACK]		CTRL & 1
[WHITE]		CTRL & 2
[RED]		CTRL & 3
[CYAN]		CTRL & 4
[PURPLE]		CTRL & 5
[GREEN]		CTRL & 6
[BLUE]		CTRL & 7
[YELLOW]		CTRL & 8
[POUND]		£
[LARROW]		←
[UPARROW]		↑
[PI]		SHIFT & ↑
[INST]		SHIFT & INST/DEL
[REV T]		see text
[Cletter]		CBM + letter
[Sletter]		SHIFT + letter

DON'T MISS OUT

Fill in your name and address and give this form to your newsagent.

Please order me a copy of YOUR COMMODORE and reserve/deliver me a copy every month.

NAME ....

ADDRESS ....

....

....

....

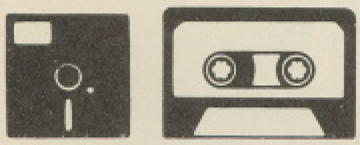
Newsagent: This magazine is made available to your wholesaler through:
S.M. Distribution Ltd
6 Leigham Court Road
Streatham
LONDON
SW16 2PG

Tel: 01-677 8111



Listings

MAKING MUSIC



PROGRAM: BOOGIE

```
37  1 REM BOOGIE
38  2 :
E1  3 PRINT CHR$(14)
CE  10 B=0:FORI=40448TO40545:REA
    DA:B=B+A:POKEI,A:NEXT
F5  12 IFB<>6215THENPRINT"TUNE E
    NTERED INCORRECTLY:CHECK DAT
    A[SPC5]CAREFULLY.":END
62  14 PRINT"TUNE 1 ENTERED CORR
    ECTLY:CHECKING TUNE 2"
1C  16 B=0:FORI=36864TO37153:REA
    DA:B=B+A:POKEI,A:NEXT
22  18 IFB<>26302THENPRINT"TUNE
    ENTERED INCORRECTLY:CHECK DA
    TA[SPC5]CAREFULLY.":END
98  20 PRINT"ALL PRESENT AND COR
    RECT : WELL DONE."
A3  25 V=54272
DD  30 POKEV+12,9:POKEV+19,9
FB  35 SYS53176
5A  40 PRINT:PRINT"[SP,SO,SK,SE]
     $[SC,SE,SD]8 TO ALTER DELAY
     (CURRENTLY SETTO 12)
9F  45 PRINT"[SP,SO,SK,SE] $[SC,
    SF]0[SC] TO ALTER W/FORM FOR
     VOICE 2
C2  50 PRINT"[SP,SO,SK,SE] $[SC,
    SF]86 TO ALTER W/FORM FOR VO
    ICE 3
B7  55 END
18  200 DATA073,004,102,005,108,
    006,053,007,146,008
85  210 DATA053,007,108,006,102,
    005,073,004,102,005
2A  220 DATA108,006,053,007,146,
    008,053,007,108,006
DC  230 DATA102,005,185,005,053,
    007,146,008,159,009
52  240 DATA113,011,159,009,146,
    008,053,007,073,004
AD  250 DATA102,005,108,006,053,
    007,146,008,053,007
65  260 DATA108,006,102,005,108,
    006,023,008,159,009
59  270 DATA205,010,216,012,205,
    010,159,009,023,008
00  280 DATA185,005,053,007,146,
    008,159,009,113,011
72  290 DATA159,009,146,008,053,
    007,255,255
C0  300 DATA075,034,149,068,149,
    068,188,064,172,057
9E  310 DATA097,051,075,034,172,
    057,097,051,198,045
B1  320 DATA052,043,177,025,198,
    045,052,043,126,038
67  330 DATA075,034,198,045,172,
    057,198,045,172,057
D5  340 DATA198,045,172,057,198,
    045,172,057,075,034
76  350 DATA052,043,075,034,052,
    043,075,034,052,043
48  360 DATA075,034,052,043,097,
    051,188,064,097,051
75  370 DATA188,064,097,051,188,
    064,097,051,188,064
43  380 DATA198,045,172,057,198,
    045,172,057,198,045
05  390 DATA172,057,198,045,172,
    057,126,038,126,038
05  400 DATA052,043,052,043,075,
    034,126,038,126,038
E2  410 DATA198,045,198,045,097,
    051,097,051,052,043
D8  420 DATA198,045,198,045,126,
    038,126,038,097,051
11  430 DATA097,051,172,057,172,
    057,198,045,097,051
63  440 DATA097,051,188,064,126,
    038,126,038,052,043
70  450 DATA052,043,075,034,126,
    038,126,038,198,045
76  460 DATA172,057,172,057,188,
    064,188,064,097,051
84  470 DATA172,057,172,057,149,
    068,097,051,097,051
EB  480 DATA172,057,172,057,198,
```

```
    045,097,051,097,051
17  490 DATA188,064,126,038,126,
    038,052,043,052,043
5C  500 DATA075,034,075,034,037,
    017,037,017,177,025
D6  510 DATA177,025,227,022,227,
    022,154,021,154,021
7D  520 DATA063,019,063,019,097,
    051,097,051,172,057
99  530 DATA172,057,198,045,198,
    045,227,022,075,034
44  540 DATA126,038,126,038,052,
    043,052,043,075,034
8C  550 DATA075,034,037,017,177,
    025,172,057,172,057
CD  560 DATA188,064,188,064,097,
    051,097,051,177,025
43  570 DATA126,038,097,051,097,
    051,172,057,172,057
37  580 DATA198,045,198,045,227,
    022,075,034,255,255
```

MAKING MUSIC

PROGRAM: INTERRUPTS

```
FA  1 REM MAYBE
38  2 :
39  3 :
E8  100 B=0
CC  102 FORI=52944TO53198
30  104 READA
AB  106 POKEI,A
0D  108 B=B+A
35  110 NEXTI
E8  112 IFB<>39215THENPRINT"DATA
     INCORRECT-CHECK CAREFULLY":
    END
63  114 PRINT"DATA ENTERED CORRE
    CTLY."
C7  120 PRINT"[CLR,DOWN2]LOAD[HO
    ME]";:POKE631,13:POKE632,82:
    POKE633,85:POKE634,78:POKE63
    5,13
1D  121 POKE198,5:END
EE  200 DATA174,207,206,232,142,
    207,206,224,012,240
3E  210 DATA004,076,049,234,234,
    162,000,142,207,206
4E  220 DATA174,206,206,234,234,
    142,206,206,234,234
49  230 DATA234,076,128,207,234,
    234,234,234,234,234
5F  240 DATA234,234,234,234,234,
    234,234,234,174,174
4B  250 DATA207,172,175,207,169,
    000,141,011,212,169
EE  260 DATA033,141,011,212,185,
    096,207,141,035,207
DC  270 DATA141,046,207,185,097,
    207,141,036,207,141
23  280 DATA047,207,189,240,144,
    201,255,240,023,141
95  290 DATA007,212,232,189,240,
    144,202,141,008,212
09  300 DATA232,232,224,240,240,
    019,142,174,207,076
C2  310 DATA049,234,162,000,142,
    174,207,142,175,207
10  320 DATA076,000,207,234,234,
    172,175,207,200,200
C2  330 DATA140,175,207,162,000,
    142,174,207,076,049
4B  340 DATA234,000,255,000,000,
    144,240,144,224,145
B6  350 DATA208,146,192,147,176,
    148,160,149,144,150
77  360 DATA128,151,112,152,096,
    153,080,154,064,155
67  370 DATA048,156,032,157,255,
    255,169,000,141,018
53  380 DATA212,169,033,141,018,
    212,174,206,206,189
30  390 DATA000,158,201,255,240,
    012,141,014,212,189
0F  400 DATA001,158,141,015,212,
    076,176,207,162,000
49  410 DATA142,206,206,076,128,
    207,000,232,232,142
16  420 DATA206,206,136,002,232,
    232,142,206,206,076
0C  430 DATA000,207,169,208,141,
    020,003,169,206,141
C1  440 DATA021,003,169,000,141,
    174,207,141,175,207
55  450 DATA141,206,206,096,000
```

BYTING 6510

FILENAME: ROM TYPEWRITER

```
10              ORG 49152
20    ;
30    CLS       EQU $E544
40    PRINTSTR  EQU $AB1E
50    PRINTNO   EQU $BDCD
60    PLOT      EQU $FFF0
70    GETIN     EQU $FFE4
80    PRINT     EQU $E716
90    ;
100   ;CLEAR THE SCREEN.
110   ;
120             JSR CLS
130   ;
140   ;PRINT HEADER.
150   ;
160             LDA #<TEXT
170             LDY #>TEXT
180             JSR PRINTSTR
190   ;
200   ;SWITCH C64 INTO BUSINESS MODE.
210   ;
220             LDA #14
230             JSR PRINT
240   ;
250   ;SET PRINT POSITION TO BEGINNING.
260   ;
270             CLC
280             LDX #6
290             LDY #0
300             JSR PLOT
310   ;
320   ;TURN CURSOR ON.
330   ;
340             LDA #0
350             STA $CC
360   ;
370   ;
380   ;
390   ;***      MAIN LOOP       ***
400   ;
410   ;GET KEYPRESS.
420   ;
430   TESTKEY   JSR GETIN
440             BEQ TESTKEY
450   ;
460   ;IF F1, EXIT FROM PROGRAM.
470   ;
480             CMP #133
490             BEQ EXIT
500   ;
510   ;IF COLUMN 40 ADJUST TO NEXT LINE
```

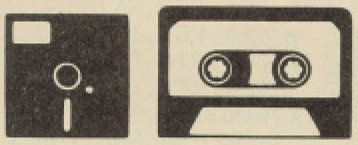
```
520  ;
530            LDX $D3
540            CPX #39
550            BNE NOADJUST
560  ;
570            PHA
580  ;
590            CLC
600            LDX $D6
610            INX
620            LDY #0
630            JSR PLOT
640  ;
650            PLA
660  ;
670  ;PRINT ASCII CHARACTER OR EXECUTE
680  ;NO-PRINTABLE CHARACTER, SUCH AS
690  ;DELETE.
700  ;
710  NOADJUST  JSR PRINT
720  ;
730  ;SAVE CURRENT PRINT POSITION.
740  ;
750            LDA $D6
760            STA 251
770            LDA $D3
780            STA 252
790  ;
800  ;PRINT CURRENT LINE NO IN HEADER.
810  ;(CURRENT LINE - 5)
820  ;
830            CLC
840            LDX #2
850            LDY #6
860            JSR PLOT
870  ;
880            SEC
890            LDA 251
900            SBC #5
910            TAX
920            LDA #0
930            JSR PRINTNO
940  ;
950  ;IF BEGINNING OF NEW LINE, BLANK
960  ;OUT COLUMN NUMBER.
970  ;
980            LDA 252
990            CMP #1
1000           BNE NOSPACES
1010 ;
1020           CLC
1030           LDX #2
1040           LDY #18
1050           JSR PLOT
1060 ;
1070           LDA #32
1080           JSR PRINT
1090           JSR PRINT
1100 ;
1110 ;PRINT CURRENT COLUMN NUMBER.
1120 ;
1130 NOSPACES  CLC
1140           LDX #2
1150           LDY #18
1160           JSR PLOT
1170 ;
1180           LDX 252
1190           LDA #0
1200           JSR PRINTNO
1210 ;
1220 ;RE-PLOT PRINT POSITION.
1230 ;
1240           CLC
1250           LDX 251
1260           LDY 252
1270           JSR PLOT
1280 ;
1290 ;BACK TO TESTKEY.
1300 ;
1310           JMP TESTKEY
1320 ;
1330 ;
1340 ;
1350 ;***    MAIN EXIT     ***
1360 ;
1370 ;RETURN TO BASIC.
1380 ;
1390 EXIT      RTS
1400 ;
1410 ;
1420 ;
1430 TEXT      BYT "                    ROM "
1440           BYT "TYPEWRITER              "
1450           BYT "        ",13
1460           BYT "LINE: 1    COLUMN:"
1470           BYT " 1                      "
1480           BYT "        ",13
1490           BYT "-------------------"
1500           BYT "-------------------"
1510           BYT "------",0
```

SHORT INTERLUDE

```
AC  100 S=49152:B=0
7B  110 FORI=0TO89:READA:B=B+A:P
    OKES+I,A:NEXTI
F4  120 IFB<>8671THENPRINT"DATA
    ERROR!":END
9F  130 PRINT"OK[DOWN2]"
A2  140 PRINT"SYS49152":PRINT"[U
    P4]"
CA  150 DATA120,169,192,141,21,3
    ,169
27  160 DATA27,141,20,3,88,96,12
    0
3D  170 DATA169,234,141,21,3,169
    ,49
A1  180 DATA141,20,3,88,96,0,169
86  190 DATA0,141,26,192,173,26,
    192
98  200 DATA10,24,105,80,141,46,
    192
22  210 DATA141,55,192,173,0,192
    ,141
99  220 DATA55,192,238,55,192,17
    3,0
7A  230 DATA192,201,0,240,5,141,
    66
21  240 DATA192,32,0,0,238,26,19
    2
62  250 DATA173,26,192,201,5,208
    ,211
E5  260 DATA76,49,234,0,0,0,0
80  270 DATA0,0,0,0,0,0
```

PROGRAM: SHORT INTERLUDE

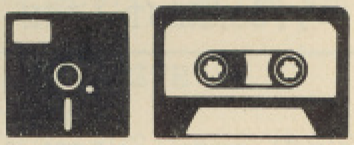
TAPE ORGANISER



```
PROGRAM: PROGRAM 1

5A  10 A=49152
98  20 READB:IFB=-1THENGOTO60
B9  30 POKEA,B:A=A+1
51  40 N=N+1:C1=C1+B:C2=C2+N*B
DD  50 GOTO20
F3  60 IFN<>825THENPRINT"WRONG N
    O OF DATA ITEMS":END
AF  70 IFC1<>109939THENPRINT"CHE
    CKSUM ERROR":END
8E  80 IFC2<>46068032THENPRINT"C
    HECKSUM ERROR":END
13  90 PRINT"CHECKSUMS OK "
D4  100 DATA169,0,133,251,169,19
    2,133,252,169,0
DD  110 DATA133,253,169,12,133,2
    54,162,4,160,0
12  120 DATA177,253,145,251,200,
    208,249,230,254,230
3E  130 DATA252,202,208,242,76,6
    3,192,169,0,133
95  140 DATA184,133,185,169,2,13
    3,186,169,8,133
D3  150 DATA171,169,1,133,172,16
    9,8,133,173,32
E2  160 DATA147,192,95,169,74,14
    1,9,3,169,192
53  170 DATA141,9,3,96,32,115,0,
    240,4,201
95  180 DATA92,240,3,76,231,167,
    32,115,0,201
E2  190 DATA83,240,11,201,76,240
    ,16,201,86,240
7E  200 DATA21,76,8,175,32,115,0
    ,32,131,192
5A  210 DATA76,174,167,32,115,0,
    32,111,193,76
A8  220 DATA174,167,32,115,0,32,
    114,193,76,174
9C  230 DATA167,162,5,134,171,32
    ,212,225,162,4
95  240 DATA181,42,149,171,202,2
    08,249,32,56,248
40  250 DATA32,143,246,32,13,193
    ,32,33,193,165
2F  260 DATA185,24,105,1,202,32,
    65,193,162,8
88  270 DATA185,172,0,32,55,193,
    162,5,200,192
5B  280 DATA5,208,243,160,0,162,
    4,177,187,195
55  290 DATA183,144,3,169,32,202
    ,32,65,193,162
99  300 DATA5,200,192,187,208,23
    7,169,2,133,171
84  310 DATA32,33,193,152,32,65,
    193,132,215,162
52  320 DATA7,177,172,32,55,193,
    162,3,230,172
1D  330 DATA208,4,230,173,202,20
    2,165,172,197,174
2E  340 DATA165,173,229,175,144,
    231,165,215,32,65
8B  350 DATA193,162,7,136,208,24
    6,200,132,192,88
EA  360 DATA24,169,0,141,160,2,7
    6,147,252,160
98  370 DATA0,132,192,173,17,208
    ,41,239,141,17
38  380 DATA208,202,208,253,136,
    208,250,120,96,160
4F  390 DATA0,169,2,32,65,193,16
    2,7,136,192
C3  400 DATA9,208,244,162,5,198,
    171,208,238,152
94  410 DATA32,65,193,162,7,136,
    208,247,202,202
89  420 DATA96,133,189,69,215,13
    3,215,169,8,133
74  430 DATA163,6,189,165,1,41,2
    47,32,98,193
CF  440 DATA162,17,9,9,32,98,193
    ,162,14,198
3E  450 DATA163,208,234,96,202,2
    08,253,144,5,162
B6  460 DATA11,202,208,253,133,1
    ,96,162,0,44
B7  470 DATA162,1,164,43,165,44,
    134,10,134,147
9C  480 DATA132,195,133,196,32,2
    12,225,32,140,193
E2  490 DATA32,122,225,76,116,16
    4,32,247,193,165
EB  500 DATA171,201,2,240,8,201,
    1,208,243,165
87  510 DATA185,240,10,173,60,3,
    133,195,173,61
84  520 DATA3,133,196,32,80,247,
    32,228,255,208
FF  530 DATA7,169,7,45,141,2,240
    ,244,32,44
4F  540 DATA168,164,183,240,11,1
    36,177,187,217,65
67  550 DATA3,209,199,152,208,24
    5,132,144,32,210
20  560 DATA245,173,62,3,56,237,
    60,3,8,24
AA  570 DATA101,195,133,174,173,
    63,3,101,196,40
3B  580 DATA237,61,3,133,175,32,
    12,194,165,189
4E  590 DATA69,215,5,144,240,4,1
    69,255,133,144
47  600 DATA76,169,245,32,69,194
    ,201,0,240,249
09  610 DATA133,171,32,115,194,1
    45,178,200,192,192
DC  620 DATA208,246,240,45,32,69
    ,194,32,115,194
32  630 DATA196,147,208,2,145,19
    5,209,195,240,2
D2  640 DATA134,144,69,215,133,2
    15,230,195,208,2
28  650 DATA230,196,165,195,197,
    174,165,196,229,175
27  660 DATA144,221,32,115,194,3
    2,13,193,200,132
2B  670 DATA192,88,24,169,0,141,
    160,2,76,147
C4  680 DATA252,32,23,248,32,13,
    193,132,215,169
00  690 DATA7,141,6,221,162,1,32
    ,131,194,38
CC  700 DATA189,165,189,201,2,20
    8,245,160,9,32
85  710 DATA115,194,201,2,240,24
    9,196,189,208,232
E5  720 DATA32,115,194,136,208,2
    46,96,169,8,133
C7  730 DATA163,32,131,194,38,18
    9,198,163,208,247
25  740 DATA165,189,96,169,16,44
    ,13,220,240,251
56  750 DATA173,13,221,142,7,221
    ,72,169,25,141
5E  760 DATA15,221,104,74,74,96,
    0,197,38,189
FF  770 DATA165,189,201,2,208,24
    6,160,9,32,115
85  780 DATA194,201,2,240,249,19
    6,189,208,233,32
65  790 DATA115,194,136,208,246,
    96,169,8,133,163
38  800 DATA32,131,194,38,189,19
    8,163,208,247,165
4D  810 DATA189,96,169,16,44,13,
    220,240,251,173
86  820 DATA13,221,142,7,221,72,
    169,25,141,15
12  830 DATA221,104,74,74,96,169
    ,11,197,46,176
12  840 DATA5,162,16,76,55,164,1
    69,0,133,251
9C  850 DATA169,192,133,252,169,
    0,133,253,169,12
EF  860 DATA133,254,162,4,160,0,
    177,251,145,253
75  870 DATA200,208,249,230,254,
    230,252,202,208,242
31  880 DATA169,1,162,1,160,1,32
    ,186,255,169
27  890 DATA9,162,48,160,195,32,
    189,255,169,0
8D  900 DATA133,251,169,8,133,25
    2,169,251,162,223
80  910 DATA160,14,32,216,255,96
    ,68,73,82,69
E8  920 DATA67,84,79,82,89,-1
```

TAPE ORGANISER

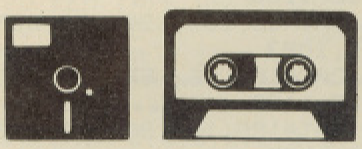


```
PROGRAM: PROGRAM 2

9B  10 POKE53281,12:POKE53280,12
    :PRINT"[CLR,BLACK]"TAB(12)"L
    IST OF PROG"
BD  20 PRINT"[DOWN]":DIMP$(19)
C8  30 FORJ=0TO19:READP$(J):PRIN
    T"[SPC3]"J")..."P$(J):NEXT
C1  40 INPUT"[DOWN]INPUT NO OF P
    ROGRAM ";PN
02  50 PRINT"YOU'VE CHOSEN [YELL
    OW]"P$(PN)" [BLACK]OK (Y/N)"
6F  60 GETG$:IFG$=""THEN60
72  70 IFG$<>"Y"THENRUN
03  80 IFPN=0THEN140
F2  90 PRINT"[CLR,DOWN2,RIGHT2]P
    RESS FAST FORWARD ON TAPE DE
    CK"
41  100 PRINT"[DOWN]THEN PRESS S
    PACE TO WIND ON TO PROGRAM"
A0  110 POKE1,39:POKE192,39:GETG
    $:IFG$<>" "THEN110
5A  120 POKE 1,7:T=TI
A9  130 IFTI<T+250*PNTHEN130
D6  140 POKE1,39:PRINT"[CLR]PRES
    S STOP ON TAPE.THE PROG IS R
    EADY"
A8  150 PRINT"[DOWN]SAVE OR LOAD
     USING £S AND £L"
9E  160 IFPEEK(3072)<>169THENPRI
    NT"ERROR":END
8C  170 SYS3072
```

```
E5  200 DATA"................."
FF  210 DATA"................."
51  220 DATA"................."
4B  230 DATA"................."
5D  240 DATA"................."
57  250 DATA"................."
28  260 DATA"................."
A2  270 DATA"................."
34  280 DATA"................."
BE  290 DATA"................."
A0  300 DATA"................."
9A  310 DATA"................."
6C  320 DATA"................."
66  330 DATA"................."
78  340 DATA"................."
F2  350 DATA"................."
C4  360 DATA"................."
DE  370 DATA"................."
B0  380 DATA"................."
AA  390 DATA"................."
```

SPLIT BAUD RATE TERMINAL

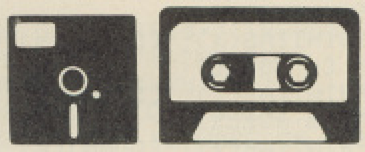


PROGRAM: C64 TERMINAL

```
10 REM SPLIT BAUD RATE TERMINAL
PROGRAM
20 REM BY W.I.SELLERS (C)1988 -
C64 VERSION
30 GOSUB1060
40 GOSUB1030
50 POKEFL,0
60 Q$="[s N]O ERRORS REPORTED"
70 PA=1:SB=1:DB=8:DU=1
80 GOSUB400
90 REM MAIN MENU
100 PRINTCHR$(14);CHR$(8);"[CLEA
R]";
110 PRINTTAB(X);"[DOWN][DOWN][s
M]AIN [s M]ENU"
120 PRINTTAB(X);"[DOWN][DOWN](1)
 [s B]AUD [s R]ATE"
130 PRINTTAB(X);"[DOWN](2) [s P]
ROTOCOL"
140 PRINTTAB(X);"[DOWN](3) [s O]
NLINE
150 PRINTTAB(X);"[DOWN](4) [s X]
MODEM [s S]END"
160 PRINTTAB(X);"[DOWN](5) [s X]
MODEM [s R]ECEIVE"
170 PRINTTAB(X);"[DOWN](6) [s P]
ET/[s A]SCII [s C]ONVERSION"
180 PRINTTAB(X);"[DOWN](7) [s Q]
UIT"
190 PRINTTAB(X);"[DOWN][DOWN][s
P]LEASE MAKE YOUR SELECTION"
200 PRINTTAB(X);"[DOWN][DOWN]";Q
$;" ";
210 GETA$:IFA$=""THEN210
220 A=VAL(A$):IFA<1ORA>7THEN210
230 Q$="[s N]O ERRORS REPORTED"
240 ONAGOSUB260,440,600,700,780,
860,990
250 GOTO90
260 REM BAUD RATE
270 PRINT"[CLEAR]";
280 IFPEEK(FL)<>0THENQ$="[s B]AU
D RATE NOT ALTERABLE":RETURN
290 PRINTTAB(X);"[DOWN][DOWN][s
B]AUD [s R]ATE"
300 PRINTTAB(X);"[DOWN][DOWN]
 [s R]ECEIVE   [s T]RANSMIT"
310 PRINTTAB(X);"[DOWN][DOWN](1)
   300        300"
320 PRINTTAB(X);"[DOWN](2)   75
      1200"
330 PRINTTAB(X);"[DOWN](3)   120
0     75"
340 PRINTTAB(X);"[DOWN](4)   120
0     1200"
350 PRINTTAB(X);"[DOWN][DOWN][s
P]LEASE MAKE YOUR SELECTION ";
360 GETA$:IFA$=""THEN360
370 A=VAL(A$):IFA<1ORA>4THEN360
380 ONAGOSUB400,410,420,430
390 RETURN
400 BR=6:POKETB,212:POKETB+1,12:
RETURN
410 BR=2:POKETB,54:POKETB+1,3:RE
TURN
420 BR=8:POKETB,80:POKETB+1,51:R
ETURN
430 BR=8:POKETB,54:POKETB+1,3:RE
TURN
440 REM PROTOCOL
450 PRINT"[CLEAR]";
460 IFPEEK(FL)<>0THENQ$="[s P]RO
TOCOL NOT ALTERABLE":RETURN
470 PRINTTAB(X);"[DOWN][DOWN][s
P]ROTOCOL"
480 INPUT"[DOWN][DOWN]          [s
 D]ATA [s B]ITS (5 TO 8)    8[LE
FT][LEFT][LEFT]";DB
490 INPUT"[DOWN][DOWN]          [s
 S]TOP [s B]ITS (1 OR 2)    1[LE
FT][LEFT][LEFT]";SB
500 PRINTTAB(X);"[DOWN][DOWN][s
P]ARITY (1) [s N]ONE"
510 PRINTTAB(X);"         (2) [s O
]DD"
520 PRINTTAB(X);"         (3) [s E
]VEN"
530 PRINTTAB(X);"         (4) [s M
]ARK"
540 PRINTTAB(X);"         (5) [s S
]PACE      ";
550 INPUT"1[LEFT][LEFT][LEFT]";P
A
560 PRINTTAB(X);"[DOWN][DOWN][s
D]UPLEX (1) [s F]ULL"
570 PRINTTAB(X);"         (2) [s H
]ALF       ";
580 INPUT"1[LEFT][LEFT][LEFT]";D
U
590 RETURN
600 REM ONLINE
610 IFPEEK(FL)<>0THEN670
620 C1$=CHR$((SB-1)*128+(8-DB)*3
2+BR)
630 IFPA<>1THENC2$=CHR$((PA-2)*6
4+(DU-1)*16))
640 IFPA=1THENC2$=CHR$((DU-1)*16
)
650 POKEFL,1:OPEN2,2,0,C1$+C2$
660 GOSUB1030
670 PRINT"[CLEAR]";
680 SYSTE
690 GOTO90
700 REM XMODEM SEND
710 IFPEEK(FL)=0THENQ$="[s R][s
S]-232 CHANNEL NOT OPEN":RETURN
720 PRINT"[CLEAR]";TAB(X);"[DOWN
][DOWN][s X]MODEM [s S]END"
730 INPUT"[DOWN][DOWN][s I]NPUT
FILE TO SEND :";F1$
740 OPEN8,8,8,F1$
750 PRINT"[CLEAR]";
760 SYSXS
770 CLOSE8:RETURN
780 REM XMODEM RECEIVE
790 IFPEEK(FL)=0THENQ$="[s R][s
S]-232 CHANNEL NOT OPEN":RETURN
800 PRINT"[CLEAR]";TAB(X);"[DOWN
][DOWN][s X]MODEM [s R]ECEIVE"
810 INPUT"[DOWN][DOWN][s I]NPUT
FILE TO RECEIVE :";F1$
820 OPEN8,8,8,F1$+",P,W"
830 PRINT"[CLEAR]";
840 SYSXR
850 CLOSE8:RETURN
860 REM ASCII CONVERSION
870 PRINT"[CLEAR]";TAB(X);"[DOWN
][DOWN][s A][s S][s C][s I][s I]
/[s P][s E][s T] [s C]ONVERSION"
880 PRINTTAB(X);"[DOWN][DOWN][s
T]O [s A]SCII OR TO [s P]ET CODE
S (A/P) ";
890 GETA$:IFA$<>"A"ANDA$<>"P"THE
N890
900 IFA$="A"THENCO=TA
910 IFA$="P"THENCO=TP
920 PRINT:INPUT"[DOWN][DOWN][s I
]NPUT READ  FILENAME :";F1$
930 OPEN8,9,9,F1$
940 INPUT"[DOWN][DOWN][s I]NPUT
WRITE FILENAME :";F2$
950 OPEN9,8,9,F2$+",P,W"
960 PRINT"[CLEAR]";
970 SYSCO
980 CLOSE8:CLOSE9:RETURN
990 REM QUIT
1000 CLOSE1
1010 SYS65412
1020 END
1030 REM SYSTEM VARIABLES
1040 TE=49152:XS=TE+3:XR=XS+3:TP
=XR+3:TA=TP+3:TB=TA+3:X=8:FL=532
47
1050 RETURN
1060 REM POKE IN MACHINE CODE
1070 PRINT"[CLEAR]POKING IN MACH
INE CODE"
1080 D=49152:L=1180
1090 T=0
1100 FORX=1TO8
1110 READA:IFA=-1THEN1160
1120 POKED,A:D=D+1:T=T+A
1130 NEXT
1140 READA:IFA=-1THEN1160:ELSEIF
T<>ATHENPRINT"ERROR IN LINE ";L:
END
1150 L=L+10:GOTO1090
1160 IFD<>50134+1THENPRINT"ADDRE
SS ERROR":END
1170 RETURN
1180 DATA76,17,192,76,223,192,76
,237,1089
1190 DATA193,76,188,194,76,196,1
94,0,1117
1200 DATA0,32,136,192,201,0,240,
9,810
1210 DATA32,106,192,32,54,192,76
,17,701
1220 DATA192,32,228,255,201,0,24
0,233,1381
1230 DATA201,171,240,9,32,71,192
,32,948
1240 DATA150,192,76,17,192,96,20
1,32,956
1250 DATA176,9,201,13,240,5,201,
20,865
1260 DATA240,1,96,32,210,255,96,
201,1131
1270 DATA20,208,3,169,127,96,201
,65,889
1280 DATA176,1,96,201,91,176,3,1
05,849
1290 DATA32,96,201,97,176,1,96,2
01,900
1300 DATA128,176,4,56,233,32,96,
41,766
```

```
1310 DATA127,96,41,127,201,127,2
08,3,930
1320 DATA169,20,96,201,65,176,1,
96,824
1330 DATA201,91,176,3,105,128,96
,201,1001
1340 DATA97,176,1,96,56,233,32,9
6,787
1350 DATA162,2,32,198,255,32,228
,255,1164
1360 DATA72,32,204,255,104,96,72
,162,997
1370 DATA2,32,201,255,104,133,15
8,32,917
1380 DATA169,192,32,204,255,96,3
2,186,1166
1390 DATA192,172,158.2.200.204.1
57,2,1087
1400 DATA240,244,140,158,2,136,1
65,158,1243
1410 DATA145,249,173,161,2,74,17
6,30,1010
1420 DATA169,16,141,14,221,173,1
5,192,941
1430 DATA141,4,221,173,16,192,14
1,5,893
1440 DATA221,169,129,32,59,239,3
2,6,887
1450 DATA239,169,17,141,14,221,9
6,169,1066
1460 DATA0,141,212,195,141,208,1
95,169,1261
1470 DATA1,141,207,195,32,228,25
5,201,1260
1480 DATA171,240,121,173,161,2,7
4,176,1118
1490 DATA243,32,87,195,32,160,19
3,176,1118
1500 DATA107,174,205,195,240,77,
201,21,1220
1510 DATA208,226,32,109,193,176,
74,169,1187
1520 DATA1,32,150,192,173,207,19
5,32,982
1530 DATA150,192,173,207,195,73,
255,32,1277
1540 DATA150,192,32,206,193,173,
206,195,1347
1550 DATA32,150,192,32,228,255,2
01,171,1261
1560 DATA240,58,173,161,2,74,176
,243,1127
1570 DATA32,87,195,32,160,193,17
6,44,919
1580 DATA174,205,195,240,14,201,
21,240,1290
1590 DATA10,201,6,208,222,238,20
7,195,1287
1600 DATA76,10,193,238,208,195,7
6,15,1011
1610 DATA193,169,4,32,150,192,32
,160,932
1620 DATA193,176,9,174,205,195,2
40,241,1433
1630 DATA201,6,208,237,96,173,21
2,195,1328
1640 DATA240,2,56,96,162,8,32,19
8,794
1650 DATA255,162,129,32,207,255,
157,214,1410
1660 DATA195,32,183,255,41,64,20
8,8,986
1670 DATA202,208,240,32,204,255,
24,96,1261
1680 DATA169,1,141,212,195,169,2
6,202,1115
1690 DATA208,241,157,214,195,76,
151,193,1435
1700 DATA169,100,141,205,195,169
,0,141,1120
1710 DATA204,195,32,136,192,72,3
2,183,1046
1720 DATA255,201,0,240,20,104,32
,228,1080
1730 DATA255,201,171,240,15,206,
204,195,1487
1740 DATA208,232,206,205,195,208
,227,24,1505
1750 DATA96,104,24,96,56,96,169,
0,641
1760 DATA141,206,195,169,128,133
,251,165,1389
1770 DATA251,189,214,195,72,32,1
50,192,1295
1780 DATA104,24,109,206,195,141,
206,195,1180
1790 DATA198,251,208,235,96,169,
21,141,1319
1800 DATA213,195,169,0,141,208,1
95,169,1290
1810 DATA1,141,207,195,32,87,195
,173,1031
1820 DATA213,195,32,150,192,32,1
60,193,1167
1830 DATA176,113,174,205,195,240
,97,201,1401
1840 DATA1,240,7,201,4,240,100,7
6,869
1850 DATA5,194,32,160,193,176,92
,174,1026
1860 DATA205,195,240,75,141,209,
195,32,1293
1870 DATA160,193,176,79,174,205,
195,240,1422
1880 DATA63,141,210,195,32,129,1
94,176,1140
1890 DATA66,174,205,195,240,50,3
2,160,1122
1900 DATA193,176,56,174,205,195,
240,40,1279
1910 DATA141,211,195,173,207,195
,205,209,1536
1920 DATA195,208,29,73,255,205,2
10,195,1370
1930 DATA208,22,173,206,195,205,
211,195,1415
1940 DATA208,14,32,168,194,238,2
07,195,1256
1950 DATA169,6,141,213,195,76,25
2,193,1245
1960 DATA238,208,195,169,21,141,
213,195,1380
1970 DATA76,252,193,169,6,32,150
,192,1070
1980 DATA96,169,0,141,206,195,16
9,128,1104
1990 DATA133,251,32,160,193,176,
23,174,1142
2000 DATA205,195,240,16,166,251,
157,214,1444
2010 DATA195,24,109,206,195,141,
206,195,1271
2020 DATA198,251,208,230,24,96,5
6,96,1159
2030 DATA162,8,32,201,255,162,12
8,189,1137
2040 DATA214,195,32,210,255,202,
208,247,1563
2050 DATA32,204,255,96,169,0,141
,211,1108
2060 DATA195,76,201,194,169,1,14
1,211,1188
2070 DATA195,169,0,141,212,195,1
41,207,1260
2080 DATA195,238,207,195,32,204,
255,32,1358
2090 DATA114,195,162,8,32,198,25
5,162,1126
2100 DATA0,32,207,255,157,214,19
5,232,1292
2110 DATA32,183,255,41,64,240,8,
169,992
2120 DATA1,141,212,195,76,251,19
4,224,1294
2130 DATA254,208,230,142,206,195
,32,204,1471
2140 DATA255,162,9,32,201,255,16
2,0,1076
2150 DATA189,214,195,232,172,211
,195,208,1616
2160 DATA10,32,106,192,201,10,24
0,9,800
2170 DATA76,30,195,32,71,192,32,
210,838
2180 DATA255,236,206,195,208,226
,173,212,1711
2190 DATA195,240,166,32,204,255,
96,72,1260
2200 DATA41,15,32,62,195,152,170
,104,771
2210 DATA41,240,74,74,74,74,201,
10,788
2220 DATA176,4,105,48,168,96,24,
105,726
2230 DATA87,158,96,32,47,195,152
,32,809
2240 DATA210,255,138,32,210,255,
96,162,1358
2250 DATA128,160,195,32,183,195,
173,207,1273
2260 DATA195,32,75,195,162,146,1
60,195,1160
2270 DATA32,183,195,173,208,195,
32,75,1093
2280 DATA195,96,162,166,160,195,
32,183,1189
2290 DATA195,173,207,195,32,75,1
95,96,1168
2300 DATA19,112,97,99,107,101,11
6,32,683
2310 DATA110,117,109,98,101,114,
32,61,742
2320 DATA32,0,13,13,13,101,114,1
14,400
2330 DATA111,114,32,32,110,117,1
09,98,723
2340 DATA101,114,32,51,32,0,19,9
8,457
2350 DATA108,111,99,107,32,110,1
17,109,793
2360 DATA98,101,114,32,61,32,0,1
34,572
2370 DATA251,132,252,160,0,177,2
51,240,1463
2380 DATA10,32,210,255,200,208,2
46,230,1391
2390 DATA252,208,242,96,0,0,0,0,
798
2400 DATA0,0,0,0,0,0,0,-1
```

SPRITE LIBRARY



PROGRAM: AIRCRAFT

```
AF  10 REM*************************
    *
4B  20 REM*    SPRITE LIRBARY
    *
A3  30 REM*    ---------------
    *
C1  40 REM*  AIRCRAFT SPRITES
```

```
     *
CE  50 REM*  BASIC DATA LOADER
     *
99  60 REM* SPRITES DESIGNED BY
     *
2E  70 REM*     MIKE BENN
     *
C5  80 REM**********************
     *
DD  90 BL=255  :LN=190  :SA=1024
    0
89  100 FOR L=0 TO BL:CX=0:FOR D
    =0 TO 15
4F  110 READ A:IF A>255THENPRINT
    "NUMBER TO LARGE";LN+(L*10):
    STOP
98  120 CX=CX+A:POKE SA+L*16+D,A
    :NEXT D
D9  130 READ A:IF A><CX THENPRIN
    T"ERROR IN LINE";LN+(L*10):S
    TOP
37  140 NEXTL:POKE43,0:POKE44,40
    :POKE45,0:POKE46,56
69  150 SAVE"AIRCRAFT",8,1:END
EF  160 REM*********************
    ***********
3B  170 REM TAPE USERS WILL NEED
     TO CHANGE          DEVICE N
    UMBER FROM 8 TO 1
FB  180 REM*********************
    ***********
2D  190 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
19  200 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
01  210 DATA 0,0,0,0,0,0,0,0,0,1
    ,0,0,0,0,0,0,1
0F  220 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,2,2
75  230 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
63  240 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
69  250 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
DE  260 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,251,251
5C  270 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
4A  280 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
5E  290 DATA 0,0,0,0,0,0,1,0,0,1
    ,0,0,7,0,0,0,9
06  300 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,255,255
A4  310 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
92  320 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
C8  330 DATA 0,0,0,0,0,0,0,0,0,0
    ,192,0,0,0,0,0,192
88  340 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,254,254
8C  350 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
FA  360 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
D0  370 DATA 0,0,0,0,0,0,1,0,0,3
    ,0,0,31,0,0,0,35
EE  380 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
D4  390 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
C2  400 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
04  410 DATA 0,0,0,0,0,0,0,128,0
    ,0,240,0,0,0,0,0,368
BE  420 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,255,255
3C  430 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
2A  440 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
5F  450 DATA 0,0,0,1,0,0,3,0,0,7
    ,0,0,255,0,0,0,266
1E  460 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,2,2
04  470 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
72  480 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
A2  490 DATA 0,0,0,0,128,0,0,192
    ,0,0,255,0,0,0,0,0,575
6E  500 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,4,4
6C  510 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
59  520 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,1,0,0,1
76  530 DATA 3,0,0,5,0,0,15,0,0,
    7,0,31,255,0,0,195,511
39  540 DATA 0.0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,253,253
8B  550 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
35  560 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,128,0,128
40  570 DATA 0,64,0,0,192,0,0,22
    4,0,0,255,240,0,131,0,0,1106
9D  580 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,5,5
83  590 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
8B  600 DATA 0,0,0,0,0,0,0,0,0,0
    ,1,0,0,1,0,0,2
AD  610 DATA 7,0,0,9,0,0,57,0,0,
    7,0,6,7,7,255,255,610
0A  620 DATA 0,6,3,0,0,0,0,0,0,0
    ,0,0,0,0,0,255,264
EB  630 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
61  640 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,192,0,192
A0  650 DATA 0,32,0,0,56,0,0,224
    ,0,0,225,128,0,255,255,192,1
    367
38  660 DATA 129,128,0,0,0,0,0,0
    ,0,0,0,0,0,0,0,1,258
33  670 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
3D  680 DATA 0,0,0,0,1,0,0,1,0,0
    ,1,0,0,1,0,0,4
7A  690 DATA 15,0,0,17,0,0,249,0
    ,0,15,0,28,15,31,247,255,872
8C  700 DATA 0,28,3,0,0,0,0,0,0,
    0,0,0,0,0,0,2,33
1E  710 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
01  720 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,224,0,224
E3  730 DATA 0,16,0,0,62,0,0,224
    ,0,0,224,224,0,255,191,240,1
    436
01  740 DATA 128,224,0,0,0,0,0,0
    ,0,0,0,0,0,0,0,251,603
63  750 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
61  760 DATA 0,1,0,0,1,0,0,1,0,0
    ,3,0,0,3,0,0,9
59  770 DATA 31,0,0,56,0,0,254,1
    5,7,31,0,253,143,0,7,127,924
AA  780 DATA 0,0,3,0,0,0,0,0,0,0
    ,0,0,0,0,0,255,258
3E  790 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,128,128
5C  800 DATA 0,0,128,0,0,0,0,0,0
    ,0,0,0,0,0,240,0,368
8F  810 DATA 0,72,0,0,88,0,0,240
    ,0,0,240,0,0,255,224,0,1119
B2  820 DATA 192,190,0,0,224,0,0
    ,0,0,0,0,0,0,0,0,1,607
92  830 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
E2  840 DATA 0,16,0,0,24,0,0,24,
    0,0,28,0,0,28,0,0,120
ED  850 DATA 255,0,1,248,0,0,124
    ,1,192,127,0,63,255,0,1,255,
    1522
FE  860 DATA 0,0,195,0,0,0,0,0,0
    ,0,0,0,0,0,0,2,197
FA  870 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
68  880 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,240,0,240
B6  890 DATA 0,72,0,0,92,0,0,252
    ,0,0,254,0,0,254,0,0,924
E7  900 DATA 231,224,0,0,0,0,0,0
    ,0,0,0,0,0,0,0,251,706
C2  910 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
F8  920 DATA 2,0,0,3,0,0,3,0,0,3
    ,0,0,3,0,0,7,21
1E  930 DATA 255,0,7,234,0,3,255
    ,0,3,255,0,1,255,0,63,255,15
    86
E2  940 DATA 0,3,152,0,0,0,0,0,0
    ,0,0,0,0,0,0,255,410
2A  950 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
78  960 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,248,0,248
A4  970 DATA 0,112,0,0,100,0,0,2
    54,0,0,255,0,0,255,0,0,976
92  980 DATA 255,224,0,1,128,0,0
    ,0,0,0,0,0,0,0,0,1,609
72  990 DATA 0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0
FD  1000 DATA 16,0,0,24,0,0,24,0
    ,0,24,0,0,24,0,0,31,143
25  1010 DATA 255,0,63,250,0,7,2
    50,0,1,255,0,0,63,0,0,127,12
    71
68  1020 DATA 0,0,3,0,0,0,0,0,0,
    0,0,0,0,0,0,2,5
59  1030 DATA 0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0
B2  1040 DATA 0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,255,192,447
AF  1050 DATA 0,173,0,0,173,160,
    0,255,248,0,255,252,0,255,25
    2,0,2023
BF  1060 DATA 255,224,0,0,0,0,0,
    0,0,0,0,0,0,0,0,251,730
A1  1070 DATA 0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0
30  1080 DATA 64,0,0,96,0,0,96,0
    ,0,112,0,0,120,0,0,127,615
72  1090 DATA 255,0,255,253,0,31
    ,253,0,7,255,0,0,63,0,0,7,13
    79
8B  1100 DATA 0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,255,255
89  1110 DATA 0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0
8F  1120 DATA 0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,255,248,503
C4  1130 DATA 0,182,128,0,182,19
    6,0,255,255,128,255,255,224,
    255,255,192,2762
A5  1140 DATA 255,248,0,0,0,0,0,
    0,0,0,0,0,0,0,0,1,504
D7  1150 DATA 0,0,3,0,0,1,0,0,0,
    0,0,0,0,0,0,0,4
DF  1160 DATA 0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0
C5  1170 DATA 0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0
CB  1180 DATA 0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,4,4
BD  1190 DATA 128,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,129
27  1200 DATA 0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0
```

```
2D 1210 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0
67 1220 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,253,253
76 1230 DATA 0,0,0,0,0,0,0,0,7,
   0,0,1,0,0,3,0,11
64 1240 DATA 0,3,0,0,3,0,0,5,0,
   0,0,0,0,0,0,0,11
75 1250 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0
EB 1260 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,251,251
AF 1270 DATA 0,0,0,0,0,0,192,0,
   0,0,0,0,128,0,0,128,448
4F 1280 DATA 0,0,128,0,0,64,0,0
   ,0,0,0,0,0,0,0,0,192
5C 1290 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0
BC 1300 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,92,92
A2 1310 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,1,255,0,256
A3 1320 DATA 0,1,0,0,3,0,0,4,0,
   0,4,0,0,6,0,0,18
99 1330 DATA 7,0,0,3,0,0,5,0,0,
   0,0,0,0,0,0,0,15
8C 1340 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,90,90
38 1350 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,255,0,0,0,255
FF 1360 DATA 0,0,128,0,0,64,0,0
   ,64,0,0,192,0,0,192,0,540
31 1370 DATA 0,128,0,0,64,0,0,0
   ,0,0,0,0,0,0,0,0,192
2C 1380 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,93,93
91 1390 DATA 0,0,0,0,0,0,0,0,0,
   0,0,1,0,63,255,0,319
87 1400 DATA 0,1,0,0,7,0,0,7,0,
   0,9,0,0,9,0,0,33
BA 1410 DATA 9,0,0,9,0,0,9,0,0,
   15,0,0,11,0,0,7,60
BC 1420 DATA 0,0,11,0,0,8,0,0,1
   6,0,0,0,0,0,0,91,126
2D 1430 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,255,252,0,0,507
E1 1440 DATA 0,0,192,0,0,192,0,
   0,128,0,0,32,0,0,32,0,576
31 1450 DATA 0,32,0,0,32,0,0,22
   4,0,0,160,0,0,192,0,0,640
EA 1460 DATA 160,0,0,32,0,0,16,
   0,0,0,0,0,0,0,0,3,211
02 1470 DATA 0,0,0,0,0,1,15,255
   ,255,0,0,1,0,0,1,0,528
4A 1480 DATA 0,7,0,0,3,0,0,7,0,
   0,15,0,0,17,0,0,49
8C 1490 DATA 17,0,0,17,0,0,25,0
   ,0,31,0,0,19,0,0,19,128
68 1500 DATA 0,0,11,0,0,15,0,0,
   16,0,0,32,0,0,64,0,138
25 1510 DATA 0,0,0,0,0,0,255,25
   5,224,0,0,0,0,0,0,192,926
83 1520 DATA 0,0,128,0,0,192,0,
   0,224,0,0,16,0,0,16,0,576
27 1530 DATA 0,16,0,0,48,0,0,24
   0,0,0,144,0,0,144,0,0,592
D2 1540 DATA 160,0,0,224,0,0,32
   ,0,0,16,0,0,8,0,0,2,442
7B 1550 DATA 0,0,0,0,0,1,0,1,25
   5,0,0,1,0,0,1,0,259
E7 1560 DATA 0,7,0,0,3,0,0,7,0,
   0,15,0,0,17,0,0,49
73 1570 DATA 17,0,0,17,0,0,25,0
   ,0,31,0,0,19,0,0,19,128
DC 1580 DATA 0,0,11,0,0,15,0,0,
   16,0,0,32,0,0,64,91,229
AC 1590 DATA 0,0,0,0,0,0,254,0,
   0,0,0,0,0,0,0,192,446
82 1600 DATA 0,0,128,0,0,192,0,
   0,224,0,0,16,0,0,16,0,576
A2 1610 DATA 0,16,0,0,48,0,0,24
   0,0,0,144,0,0,144,0,0,592
E2 1620 DATA 160,0,0,224,0,0,32
   ,0,0,16,0,0,8,0,0,3,443
61 1630 DATA 0,0,0,0,0,1,15,255
   ,255,0,0,1,0,0,1,0,528
49 1640 DATA 0,7,0,0,67,0,0,127
   ,0,0,127,0,0,57,0,0,385
C2 1650 DATA 57,0,0,56,0,0,24,0
   ,0,31,0,0,30,0,0,30,228
65 1660 DATA 0,0,15,0,0,7,0,0,2
   0,0,0,12,0,0,6,2,62
C4 1670 DATA 0,0,0,0,0,0,255,25
   5,224,0,0,0,0,0,0,192,926
32 1680 DATA 0,0,128,0,0,192,0,
   0,224,0,0,16,0,0,16,0,576
77 1690 DATA 0,144,0,0,176,0,0,
   240,0,0,80,0,0,80,0,0,720
AC 1700 DATA 224,0,0,240,0,0,8,
   0,0,12,0,0,4,0,0,254,742
71 1710 DATA 0,0,0,0,0,1,15,255
   ,255,0,0,1,0,0,1,0,528
7C 1720 DATA 4,7,0,7,195,0,7,25
   5,0,5,255,0,0,246,0,0,981
0C 1730 DATA 244,0,0,116,0,0,12
   4,0,0,63,0,0,62,0,0,30,639
38 1740 DATA 0,0,31,0,0,103,0,0
   ,25,0,0,7,0,0,0,89,255
94 1750 DATA 0,0,0,0,0,0,255,25
   5,224,0,0,0,0,0,0,192,926
D9 1760 DATA 0,0,128,0,0,192,0,
   0,192,0,0,224,0,0,80,0,816
2C 1770 DATA 0,40,0,0,24,0,0,25
   2,0,0,60,0,0,124,0,0,500
86 1780 DATA 248,0,0,240,0,0,8,
   0,0,4,0,0,0,0,0,255,755
39 1790 DATA 0,0,0,0,0,0,3,255,
   255,0,0,0,0,0,0,0,513
C6 1800 DATA 128,7,1,255,131,1,
   255,255,0,131,255,0,1,252,0,
   0,1672
1F 1810 DATA 252,0,0,124,0,0,62
   ,0,0,31,0,0,15,0,0,7,491
EF 1820 DATA 0,0,23,0,0,25,0,0,
   6,0,0,1,0,0,0,89,144
A2 1830 DATA 0,0,0,32,0,0,255,2
   55,252,32,0,0,32,0,0,252,111
   0
14 1840 DATA 0,0,248,0,0,252,0,
   0,254,0,0,66,0,0,65,0,885
47 1850 DATA 0,64,128,0,65,128,
   0,255,192,0,240,224,0,241,22
   4,0,1761
B7 1860 DATA 255,192,0,254,0,0,
   40,0,0,231,0,0,0,0,0,255,122
   7
4E 1870 DATA 0,0,0,0,0,0,3,255,
   255,0,0,0,0,0,0,32,545
D5 1880 DATA 0,7,255,255,131,12
   7,255,255,64,127,255,0,15,25
   2,0,3,2001
DE 1890 DATA 252,0,1,252,0,0,25
   4,0,0,127,0,0,63,0,0,15,964
DF 1900 DATA 0,0,7,0,0,3,0,0,2,
   0,0,2,0,0,63,2,79
90 1910 DATA 32,0,0,48,0,0,255,
   255,255,48,0,0,48,0,0,255,11
   96
79 1920 DATA 0,0,254,0,0,255,12
   8,0,255,192,0,48,160,0,48,14
   4,1484
FD 1930 DATA 0,48,72,0,48,68,0,
   255,254,0,255,195,128,255,19
   9,0,1777
87 1940 DATA 255,222,0,255,248,
   0,2,0,0,2,96,0,255,128,0,254
   ,1717
0F 1950 DATA 0,0,0,0,0,0,0,0,25
   5,0,0,0,0,0,0,128,383
BA 1960 DATA 0,7,255,255,131,12
   7,255,255,32,127,255,0,15,25
   2,0,3,1969
AE 1970 DATA 252,0,1,252,0,0,25
   4,0,0,127,0,0,63,0,0,15,964
E3 1980 DATA 0,0,7,0,0,3,0,0,2,
   0,0,2,0,0,63,0,77
3A 1990 DATA 32,0,0,48,0,0,255,
   252,0,48,0,0,48,0,0,255,938
E9 2000 DATA 0,0,254,0,0,255,12
   8,0,255,192,0,48,160,0,48,14
   4,1484
6D 2010 DATA 0,48,72,0,48,68,0,
   255,254,0,255,195,128,255,19
   9,0,1777
F4 2020 DATA 255,222,0,255,248,
   0,2,0,0,2,96,0,255,128,0,1,1
   464
64 2030 DATA 0,0,0,0,0,0,0,0,1,
   0,0,1,0,0,1,0,3
72 2040 DATA 0,1,0,3,225,0,15,2
   55,0,63,255,0,255,255,3,255,
   1585
1A 2050 DATA 255,0,0,31,0,0,5,0
   ,0,9,0,0,11,0,0,15,326
A3 2060 DATA 0,0,3,0,0,0,0,0,0,
   0,0,0,0,0,0,91,94
4D 2070 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0
94 2080 DATA 0,0,15,128,0,255,2
   24,0,255,248,0,255,254,0,255
   ,255,2144
9C 2090 DATA 128,240,0,0,64,0,0
   ,160,0,0,160,0,0,224,0,0,976
FA 2100 DATA 192,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,91,283
97 2110 DATA 0,0,0,0,0,0,0,0,1,
   0,0,1,0,0,1,0,3
61 2120 DATA 15,225,0,63,255,0,
   127,255,0,127,255,1,255,255,
   3,255,2091
27 2130 DATA 255,0,0,1,0,0,3,0,
   0,3,0,0,31,0,0,15,308
1B 2140 DATA 0,0,7,0,0,0,0,0,0,
   0,0,0,0,0,0,2,9
FD 2150 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0
0B 2160 DATA 0,0,15,224,0,127,2
   48,0,255,254,0,255,240,0,255
   ,0,1873
D1 2170 DATA 0,240,0,0,224,0,0,
   0,0,0,224,0,0,224,0,0,912
45 2180 DATA 224,0,0,224,0,0,0,
   0,0,0,0,0,0,0,0,164,612
C7 2190 DATA 0,0,0,0,0,0,0,0,1,
   0,0,1,0,0,1,0,3
84 2200 DATA 0,1,1,255,255,0,25
   5,255,0,127,255,0,63,255,0,6
   3,1785
77 2210 DATA 255,0,31,255,0,15,
   252,0,0,0,0,0,0,0,0,15,823
47 2220 DATA 0,0,7,0,0,0,0,0,0,
   0,0,0,0,0,0,0,7
2D 2230 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0
5B 2240 DATA 0,0,255,254,0,255,
   254,0,255,254,0,255,248,0,25
   5,128,2413
CB 2250 DATA 0,240,0,0,112,0,0,
   52,0,0,124,0,0,248,0,0,776
42 2260 DATA 248,0,0,56,0,0,0,0
   ,0,0,0,0,0,0,0,89,393
77 2270 DATA 0,0,0,0,0,0,0,0,1,
   0,0,1,0,0,1,0,3
43 2280 DATA 0,1,255,255,255,12
   7,255,255,31,255,255,15,255,
   255,15,255,2739
05 2290 DATA 255,7,255,255,7,25
   5,252,7,254,0,0,0,0,0,0,63,1
   610
73 2300 DATA 0,0,63,0,0,32,0,0,
   0,0,0,0,0,0,0,89,184
5C 2310 DATA 0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0
BC 2320 DATA 0,0,255,255,254,25
   5,255,248,255,255,0,255,248,
   0,255,128,2918
```

```
71   2330 DATA 0,240,0,0,64,0,0,3
     5,0,0,127,0,0,252,0,0,718
B1   2340 DATA 248,0,0,8,0,0,0,0,
     0,0,0,0,0,0,0,165,421
A0   2350 DATA 0,0,0,0,0,0,0,0,2,
     0,0,2,0,0,2,0,6
5B   2360 DATA 0,1,0,0,1,0,0,15,0
     ,0,255,0,15,255,0,127,669
57   2370 DATA 255,3,255,255,63,2
     55,254,63,255,248,31,255,240
     ,3,255,255,2945
5B   2380 DATA 0,254,63,0,0,32,0,
     0,0,0,0,0,0,0,0,89,438
1A   2390 DATA 0,0,0,0,0,0,0,0,25
     4,0,15,248,1,255,192,31,996
60   2400 DATA 254,0,255,248,0,25
     5,192,0,255,0,0,240,0,0,224,
     0,1923
40   2410 DATA 0,224,0,0,19,0,0,1
     5,0,0,124,0,0,248,0,0,630
8A   2420 DATA 136,0,0,0,0,0,0,0,
     0,0,0,0,0,0,0,252,388
DE   2430 DATA 0,0,2,0,0,2,0,0,2,
     0,0,1,0,0,1,0,8
BC   2440 DATA 0,15,0,0,255,0,15,
     255,0,127,255,3,255,255,63,2
     55,1753
C7   2450 DATA 254,63,255,248,31,
     255,240,3,255,241,0,254,1,0,
     0,1,2101
9A   2460 DATA 0,0,1,0,0,3,0,0,1,
     0,0,0,0,0,0,95,100
01   2470 DATA 0,0,254,0,15,248,1
     ,255,192,31,254,0,255,248,0,
     255,2008
25   2480 DATA 192,0,255,0,0,240,
     0,0,224,0,0,224,0,0,19,0,115
     4
B4   2490 DATA 0,15,0,0,124,0,0,2
     48,0,0,200,0,0,192,0,0,779
3B   2500 DATA 64,0,0,128,0,0,0,0
     ,0,0,0,0,0,0,0,90,282
01   2510 DATA 0,0,1,0,0,1,0,0,1,
     0,0,1,255,255,255,127,896
AC   2520 DATA 255,255,31,255,255
     ,15,255,255,15,255,255,7,255
     ,255,7,255,2880
06   2530 DATA 252,7,254,0,0,0,0,
     0,0,0,0,0,0,0,0,0,513
82   2540 DATA 0,0,0,0,0,0,0,0,0,
     0,0,1,0,0,1,91,93
3B   2550 DATA 0,0,0,0,0,0,0,0,0,
     0,0,0,255,255,254,255,1019
4B   2560 DATA 255,248,255,255,0,
     255,248,0,255,128,0,252,0,0,
     76,0,2227
40   2570 DATA 0,24,0,0,56,0,0,56
     ,0,0,120,0,0,104,0,0,360
B1   2580 DATA 112,0,0,48,0,0,96,
     0,0,160,0,0,48,0,0,1,465
0A   2590 DATA 0,0,0,0,0,0,0,0,0,
     0,0,0,15,0,0,7,22
F2   2600 DATA 224,0,3,252,0,1,25
     5,128,1,255,240,0,255,254,0,
     63,1931
FB   2610 DATA 255,0,31,255,0,7,2
     55,0,7,255,0,1,255,0,0,63,13
     84
64   2620 DATA 0,0,40,0,0,36,0,0,
     36,0,0,54,0,0,0,6,172
83   2630 DATA 0,0,0,0,0,0,0,0,0,
     0,0,0,0,0,0,0,0
88   2640 DATA 0,0,9,0,0,8,0,0,16
     ,0,0,16,0,0,224,0,272
FC   2650 DATA 0,248,0,0,255,128,
     0,255,240,0,255,254,0,255,25
     5,192,2337
0B   2660 DATA 0,15,248,0,0,31,0,
     0,0,0,0,0,0,0,0,253,547
50   2670 DATA 0,0,0,0,0,0,0,0,0,
     15,0,0,7,224,0,3,249
10   2680 DATA 252,0,1,255,128,1,
     255,240,0,255,254,0,63,255,0
     ,31,1990
9B   2690 DATA 255,0,7,255,0,7,25
     5,0,1,255,0,0,63,0,0,7,1105
C3   2700 DATA 0,0,0,0,0,0,0,0,1,
     0,0,1,0,0,1,6,9
33   2710 DATA 0,0,0,0,0,0,0,0,0,
     0,0,0,0,0,0,0,0
13   2720 DATA 0,0,4,0,0,4,0,0,8,
     0,0,200,0,0,248,0,464
C4   2730 DATA 0,252,0,0,255,0,0,
     255,192,0,255,240,0,255,252,
     0,1956
44   2740 DATA 255,255,0,129,255,
     192,128,15,240,0,0,252,0,0,0
     ,0,1721
```

SPRITE LIBRARY

AIRCRAFT DISPLAY

```
85   10 REM***************************
     ****
31   20 REM* SPRITE LIRBARY DISPL
     AY *
E3   30 REM*        AIRCRAFT
        *
CB   40 REM***************************
     ****
6F   50 POKE55,0:POKE56,40:X=X+1:
     IFX=1THENLOAD"AIRCRAFT",8,1
5A   60 V=53248:PRINT"[CLS][BLU][
     26CD][9CR][REV]F7 TO STOP AN
     IMATION"
86   70 POKEV+21,15:POKEV+23,12:P
     OKEV+29,12:POKEV+32,1:POKEV+
     33,1:POKEV+39,0
6D   80 POKEV+40,0:POKEV+41,0:POK
     EV+42,0:POKEV,80:POKEV+1,150
     :POKEV+2,104:POKEV+3,150
9C   90 POKEV+4,200:POKEV+5,150:P
     OKEV+6,248:POKEV+7,150
8D   100 INPUT"[HOM][CD]START SPR
     ITE";S:INPUT"END SPRITE";E:I
     NPUT"DELAY";D
05   110 FORSP=STOESTEP2:FORT=0TO
     D:NEXT:PRINT"[HOM]"TAB(23)"S
     PRITE NO.=";SP:POKE2040,SP
37   120 POKE2041,SP+1:POKE2042,S
     P:POKE2043,SP+1:NEXT:GETK$:I
     FK$="[F7]"THEN100
3D   130 GOTO110
```

EDUCATING SYDNEY

PROGRAM: SID TEST

```
AE   5 REM SID TEST PROGRAM
96   10 SID=54272:REM CHANGE TO (
     BASE+49)
D4   20 POKESID+0,00:POKESID+1,0:
      REM PITCH
F1   30 POKESID+4,PEEK(SID+4)OR32
     :REM SHAPE
F8   40 POKESID+5,PEEK(SID+5)OR10
     :REM DECAY
D3   50 POKESID+5,PEEK(SID+5)OR80
     :REM ATTACK
9A   60 POKESID+6,PEEK(SID+6)OR24
     0:REM SUST
9B   70 POKESID+6,PEEK(SID+6)OR5:
      REM REL
6C   80 POKESID+24,PEEK(SID+24)OR
     15:REM VOL
BC   90 POKESID+4,PEEK(SID+4)OR1:
     REM GATE ON
01   100 FORD=1TO255:POKESID+1,PE
     EK(SID+1)+1
30   110 NEXTD
E0   120 POKESID+4,PEEK(SID+4)AND
     254:REM OFF
48   200 END
```

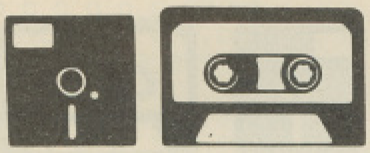
EDUCATING SYDNEY

PROGRAM: SIDREAD

```
57   10 REM READ/WRITE SID TABLE
2E   20 :
2A   30 REM SET BASE VALUE IN LIN
     E 60
08   50 :
D6   60 BASE=49152:REM DEFAULT LO
     CATION $C000
D1   70 FORA=0TO48:READC:POKEBASE
     +A,C:NEXTA
4C   80 H=(INT(BASE/255)):L=BASE-
     (H*256)
9A   90 POKEBASE+5,L+49
0E   100 POKEBASE+6,H
C9   110 POKEBASE+15,L+47
63   120 POKEBASE+16,H
5B   130 POKEBASE+21,L+48
7C   140 POKEBASE+22,H
9C   150 POKEBASE+24,L+35
F7   160 POKEBASE+29,H
AA   170 POKEBASE+38,L+49
4C   180 POKEBASE+39,H
69   200 SYSBASE
7D   210 PRINT"TABLE CLEARED AND
     INITIALISED"
FC   220 PRINT:PRINT"SYS"BASE" TO
      RE-INITIALISE"
47   230 PRINT:PRINT"SYS"BASE+10"
      TO WARM START."
F2   240 PRINT:PRINT"TABLE STARTS
      AT "BASE+49
A0   300 PRINT"SAVE? (DISK ONLY)
     Y/N"
88   310 GETK$:IFK$=""THEN310
```

```
77  320 IFK$<>"Y"THENEND
A9  400 INPUT"FILENAME? (MAX 16 CHRS)";F$
02  410 PRINT"PRESS A KEY WHEN READY"
5E  420 GETK$:IFK$=""THEN420
18  430 OPEN1,8,1,F$
9E  440 PRINT#1,CHR$(L);CHR$(H);
50  450 FORI=BASETOBASE+50
45  460 PRINT#1,CHR$(PEEK(I));
9E  470 NEXTI
72  480 CLOSE1
B4  490 PRINT"DONE"
75  500 END
9D  1000 DATA169,0,162,24,157,49,192,202
17  1100 DATA16,250,120,173,20,3,141,47
A0  1200 DATA192,173,21,3,141,48,192,169
EC  1300 DATA35,141,20,3,169,192,141,21
D4  1400 DATA3,88,96,162,24,189,49,192
5A  1500 DATA157,0,212,202,16,247,76,49
F8  1600 DATA234
```

SPLIT BAUD RATE TERMINAL



PROGRAM: C128 TERMINAL

```
10 REM SPLIT BAUD RATE TERMINAL PROGRAM
20 REM BY W.I.SELLERS (C)1988
30 BANK15
40 GOSUB1070
50 IFPEEK(215)AND128THENFAST
60 Q$="[s N]O ERRORS REPORTED"
70 PA=1:SB=1:DB=8:DU=1:FL=0:X=8
80 TE=4864:XS=TE+3:XR=XS+3:TP=XR+3:TA=TP+3:TB=TA+3
90 GOSUB420
100 REM MAIN MENU
110 IFPEEK(215)AND128THENSYS52591
120 PRINTCHR$(14);CHR$(8);"[CLEAR]";
130 PRINTTAB(X);"[DOWN][DOWN][s M]AIN [s M]ENU"
140 PRINTTAB(X);"[DOWN][DOWN](1) [s B]AUD [s R]ATE"
150 PRINTTAB(X);"[DOWN](2) [s P]ROTOCOL"
160 PRINTTAB(X);"[DOWN](3) [s O]NLINE"
170 PRINTTAB(X);"[DOWN](4) [s X]MODEM [s S]END"
180 PRINTTAB(X);"[DOWN](5) [s X]MODEM [s R]ECEIVE"
190 PRINTTAB(X);"[DOWN](6) [s P]ET/[s A]SCII [s C]ONVERSION"
200 PRINTTAB(X);"[DOWN](7) [s Q]UIT"
210 PRINTTAB(X);"[DOWN][DOWN][s P]LEASE MAKE YOUR SELECTION"
220 PRINTTAB(X);"[DOWN][DOWN]";Q$;" ";
230 GETA$:IFA$=""THEN230
240 A=VAL(A$):IFA<1ORA>7THEN230
250 Q$="[s N]O ERRORS REPORTED"
260 ONAGOSUB280,460,620,710,800,890,1030
270 GOTO100
280 REM BAUD RATE
290 SCNCLR
300 IFFL<>0THENQ$="[s B]AUD RATE NOT ALTERABLE":RETURN
310 PRINTTAB(X);"[DOWN][DOWN][s B]AUD [s R]ATE"
320 PRINTTAB(X);"[DOWN][DOWN] [s R]ECEIVE   [s T]RANSMIT"
330 PRINTTAB(X);"[DOWN][DOWN](1)   300       300"
340 PRINTTAB(X);"[DOWN](2)    75      1200"
350 PRINTTAB(X);"[DOWN](3)    1200      75"
360 PRINTTAB(X);"[DOWN](4)    1200      1200"
370 PRINTTAB(X);"[DOWN][DOWN][s P]LEASE MAKE YOUR SELECTION ";
380 GETA$:IFA$=""THEN380
390 A=VAL(A$):IFA<1ORA>4THEN380
400 ONAGOSUB420,430,440,450
410 RETURN
420 BR=6:POKETB,212:POKETB+1,12:RETURN
430 BR=2:POKETB,54:POKETB+1,3:RETURN
440 BR=8:POKETB,80:POKETB+1,51:RETURN
450 BR=8:POKETB,54:POKETB+1,3:RETURN
460 REM PROTOCOL
470 SCNCLR
480 IFFL<>0THENQ$="[s P]ROTOCOL NOT ALTERABLE":RETURN
490 PRINTTAB(X);"[DOWN][DOWN][s P]ROTOCOL"
500 INPUT"[DOWN][DOWN]          [s D]ATA [s B]ITS (5 TO 8)    8[LEFT][LEFT][LEFT]";DB
510 INPUT"[DOWN][DOWN]          [s S]TOP [s B]ITS (1 OR 2)    1[LEFT][LEFT][LEFT]";SB
520 PRINTTAB(X);"[DOWN][DOWN][s P]ARITY (1) [s N]ONE"
530 PRINTTAB(X);"         (2) [s O]DD"
540 PRINTTAB(X);"         (3) [s E]VEN"
550 PRINTTAB(X);"         (4) [s M]ARK"
560 PRINTTAB(X);"         (5) [s S]PACE       ";
570 INPUT"1[LEFT][LEFT][LEFT]";PA
580 PRINTTAB(X);"[DOWN][DOWN][s D]UPLEX (1) [s F]ULL"
590 PRINTTAB(X);"         (2) [s H]ALF       ";
600 INPUT"1[LEFT][LEFT][LEFT]";DU
610 RETURN
620 REM ONLINE
630 IFFL<>0THEN680
640 C1$=CHR$((SB-1)*128+(8-DB)*32+BR)
650 IFPA<>1THENC2$=CHR$((PA-2)*64+(DU-1)*16))
660 IFPA=1THENC2$=CHR$((DU-1)*16)
670 OPEN2,2,0,C1$+C2$:FL=1
680 SCNCLR
690 SYSTE
700 GOTO100
710 REM XMODEM SEND
720 IFFL=0THENQ$="[s R][s S]-232 CHANNEL NOT OPEN":RETURN
730 PRINT"[CLEAR]";TAB(X);"[DOWN][DOWN][s X]MODEM [s S]END"
740 INPUT"[DOWN][DOWN][s I]NPUT FILE TO SEND :";F1$
750 OPEN8,8,8,F1$
760 IFDS<>0THENQ$=DS$:CLOSE8:RETURN
770 SCNCLR
780 SYSXS
790 CLOSE8:Q$=DS$:RETURN
800 REM XMODEM RECEIVE
810 IFFL=0THENQ$="[s R][s S]-232 CHANNEL NOT OPEN":RETURN
820 PRINT"[CLEAR]";TAB(X);"[DOWN][DOWN][s X]MODEM [s R]ECEIVE"
830 INPUT"[DOWN][DOWN][s I]NPUT FILE TO RECEIVE :";F1$
840 OPEN8,8,8,F1$+",P,W"
850 IFDS<>0THENQ$=DS$:CLOSE8:RETURN
860 SCNCLR
870 SYSXR
880 CLOSE8:Q$=DS$:RETURN
890 REM ASCII CONVERSION
900 PRINT"[CLEAR]";TAB(X);"[DOWN][DOWN][s A][s S][s C][s I][s I]/[s P][s E][s T] [s C]ONVERSION"
910 PRINTTAB(X);"[DOWN][DOWN][s T]O [s A]SCII OR TO [s P]ET CODES (A/P) ";
920 GETA$:IFA$<>"A"ANDA$<>"P"THEN920
930 IFA$="A"THENCO=TA:ELSECO=TP
940 PRINT:INPUT"[DOWN][DOWN][s I]NPUT READ  FILENAME :";F1$
950 OPEN8,8,8,F1$
960 IFDS<>0THENQ$=DS$:CLOSE8:RETURN
970 INPUT"[DOWN][DOWN][s I]NPUT WRITE FILENAME :";F2$
980 OPEN9,8,9,F2$+",P,W"
990 IFDS<>0THENQ$=DS$:CLOSE8:CLOSE9:RETURN
1000 SCNCLR
1010 SYSCO
1020 CLOSE8:CLOSE9:Q$=DS$:RETURN
1030 REM QUIT
1040 CLOSE1
1050 SYS65412
1060 END
1070 REM POKE IN MACHINE CODE
1080 PRINT"[CLEAR]POKING IN MACHINE CODE"
1090 D=4864:L=1190
1100 T=0
1110 FORX=1TO8
1120 READA:IFA=-1THEN1170
1130 POKED,A:D=D+1:T=T+A
1140 NEXT
1150 READA:IFA=-1THEN1170:ELSEIFT<>ATHENPRINT"ERROR IN LINE ";L:END
1160 L=L+10:GOTO1100
1170 IFD<>5845+1THENPRINT"ADDRESS ERROR":END
1180 RETURN
1190 DATA76,17,19,76,223,19,76,237,743
1200 DATA20,76,188,21,76,196,21,0,598
1210 DATA0,32,136,19,201,0,240,9,637
1220 DATA32,106,19,32,54,19,76,17,355
1230 DATA19,32,228,255,201,0,240,233,1208
1240 DATA201,171,240,9,32,71,19,32,775
1250 DATA150,19,76,17,19,96,201,32,610
```

```
1260 DATA176,9,201,13,240,5,201,
20,865
1270 DATA240,1,96,32,210,255,96,
201,1131
1280 DATA20,208,3,169,127,96,201
,65,889
1290 DATA176,1,96,201,91,176,3,1
05,849
1300 DATA32,96,201,97,176,1,96,2
01,900
1310 DATA128,176,4,56,233,32,96,
41,766
1320 DATA127,95,41,127,201,127,2
08,3,930
1330 DATA169,20,95,201,65,176,1,
96,824
1340 DATA201,91,176,3,105,129,96
,201,1001
1350 DATA97,176,1,95,56,233,32,9
6,787
1360 DATA162,2,32,198,255,32,228
,255,1164
1370 DATA72,32,204,255,104,96,72
,162,997
1380 DATA2,32,201,255,104,133,15
8,32,917
1390 DATA169,19,32,204,255,96,32
,186,993
1400 DATA19,172,27,10,200,204,26
,10,668
1410 DATA240,244,140,27,10,136,1
65,158,1120
1420 DATA145,202,173,15,10,74,17
6,30,825
1430 DATA169,16,141,14,221,173,1
5,19,768
1440 DATA141,4,221,173,16,19,141
,5,720
1450 DATA221,169,129,32,127,230,
32,74,1014
1460 DATA230,169,17,141,14,221,9
6,169,1057
1470 DATA0,141,212,22,141,208,22
,169,915
1480 DATA1,141,207,22,32,228,255
,201,1087
1490 DATA171,240,121,173,15,10,7
4,176,980
1500 DATA243,32,87,22,32,160,20,
176,772
1510 DATA107,174,205,22,240,77,2
01,21,1047
1520 DATA208,225,32,109,20,176,7
4,169,1014
1530 DATA1,32,150,19,173,207,22,
32,636
1540 DATA150,19,173,207,22,73,25
5,32,931
1550 DATA150,19,32,206,20,173,20
6,22,828
1560 DATA32,150,19,32,228,255,20
1,171,1088
1570 DATA240,58,173,15,10,74,176
,243,989
1580 DATA32,87,22,32,160,20,176,
44,573
1590 DATA174,205,22,240,14,201,2
1,240,1117
1600 DATA10,201,6,208,222,238,20
7,22,1114
1610 DATA76,10,20,238,208,22,76,
15,665
1620 DATA20,169,4,32,150,19,32,1
60,586
1630 DATA20,176,9,174,205,22,240
,241,1087
1640 DATA201,6,208,237,96,173,21
2,22,1155
1650 DATA240,2,56,96,162,8,32,19
8,794
1660 DATA255,162,128,32,207,255,
157,214,1410
1670 DATA22,32,183,255,41,64,208
,8;813
1680 DATA202,208,240,32,204,255,
24,96,1261
1690 DATA169,1,141,212,22,169,25
,202,942
1700 DATA208,241,157,214,22,76,1
51,20,1089
1710 DATA169,100,141,205,22,169,
0,141,947
1720 DATA204,22,32,136,19,72,32,
183,700
1730 DATA255,201,0,240,20,104,32
,228,1080
1740 DATA255,201,171,240,15,206,
204,22,1314
1750 DATA208,232,206,205,22,208,
227,24,1332
1760 DATA96,104,24,96,56,96,169,
0,641
1770 DATA141,206,22,169,128,133,
251,166,1216
1780 DATA251,189,214,22,72,32,15
0,19,949
1790 DATA104,24,109,206,22,141,2
06,22,834
1800 DATA198,251,208,235,96,169,
21,141,1319
1810 DATA213,22,169,0,141,208,22
,169,944
1820 DATA1,141,207,22,32,87,22,1
73,685
1830 DATA213,22,32,150,19,32,160
,20,648
1840 DATA176,113,174,205,22,240,
97,201,1228
1850 DATA1,240,7,201,4,240,100,7
6,869
1860 DATA5,21,32,160,20,176,92,1
74,680
1870 DATA205,22,240,76,141,209,2
2,32,947
1880 DATA160,20,176,79,174,205,2
2,240,1076
1890 DATA63,141,210,22,32,129,21
,176,794
1900 DATA66,174,205,22,240,50,32
,160,949
1910 DATA20,176,56,174,205,22,24
0,40,933
1920 DATA141,211,22,173,207,22,2
05,209,1190
1930 DATA22,208,29,73,255,205,21
0,22,1024
1940 DATA208,22,173,206,22,205,2
11,22,1069
1950 DATA208,14,32,158,21,238,20
7,22,910
1960 DATA169,5,141,213,22,76,252
,20,899
1970 DATA238,208,22,169,21,141,2
13,22,1034
1980 DATA76,252,20,169,6,32,150,
19,724
1990 DATA96,169,0,141,206,22,169
,128,931
2000 DATA133,251,32,160,20,176,2
3,174,969
2010 DATA205,22,240,16,166,251,1
57,214,1271
2020 DATA22,24,109,206,22,141,20
6,22,752
2030 DATA198,251,208,230,24,96,5
6,96,1159
2040 DATA162,8,'32,201,255,162,12
8,189,1137
2050 DATA214,22,32,210,255,202,2
08,247,1390
2060 DATA32,204,255,96,169,0,141
,211,1108
2070 DATA22,76,201,21,169,1,141,
211,842
2080 DATA22,169,0,141,212,22,141
,207,914
2090 DATA22,238,207,22,32,204,25
5,32,1012
2100 DATA114,22,162,8,32,198,255
,162,953
2110 DATA0,32,207,255,157,214,22
,232,1119
2120 DATA32,183,255,41,64,240,8,
169,992
2130 DATA1,141,212,22,76,251,21,
224,948
2140 DATA254,208,230,142,206,22,
32,204,1298
2150 DATA255,162,9,32,201,255,16
2,0,1076
2160 DATA189,214,22,232,172,211,
22,208,1270
2170 DATA10,32,106,19,201,10,240
,9,627
2180 DATA76,30,22,32,71,19,32,21
0,492
2190 DATA255,236,206,22,208,225,
173,212,1538
2200 DATA22,240,166,32,204,255,9
6,72,1087
2210 DATA41,15,32,62,22,152,170,
104,598
2220 DATA41,240,74,74,74,74,201,
10,788
2230 DATA176,4,105,48,168,96,24,
105,726
2240 DATA87,168,96,32,47,22,152,
32,636
2250 DATA210,255,138,32,210,255,
96,162,1358
2260 DATA128,160,22,32,183,22,17
3,207,927
2270 DATA22,32,75,22,162,146,160
,22,641
2280 DATA32,183,22,173,208,22,32
,75,747
2290 DATA22,96,162,166,160,22,32
,183,843
2300 DATA22,173,207,22,32,75,22,
96,649
2310 DATA19,112,97,99,107,101,11
6,32,683
2320 DATA110,117,109,98,101,114,
32,61,742
2330 DATA32,0,13,13,13,101,114,1
14,400
2340 DATA111,114,32,32,110,117,1
09,98,723
2350 DATA101,114,32,61,32,0,19,9
8,457
2360 DATA108,111,99,107,32,110,1
17,109,793
2370 DATA98,101,114,32,61,32,0,1
34,572
2380 DATA251,132,252,160,0,177,2
51,240,1463
2390 DATA10,32,210,255,200,208,2
46,230,1391
2400 DATA252,208,242,96,0,0,0,0,
798
2410 DATA0,0,0,0,0,0.0,-1
```

SIMPLE

PROGRAM: SIMPLE LOADER

```
10 SCNCLR:IF PEEK(44)<>40 THEN P
RINT"IMPORTANT - SAVE PROGRAM, A
ND ENTER POKES":STOP
20 FOR A=4096 TO 4155:READ D:POK
E A,D:C=C+D:NEXT.A
30 PRINT"SECTION 1 ";:IF C=11524
 THEN PRINT"OK":ELSEPRINT"ERROR:
"ABS(11524-C):END
40 FOR A=5204 TO 6194:READ D:POK
E A,D:C=C+D:NEXT A
50 PRINT"SECTION 2 ";:IF C=14763
7 THEN PRINT"OK":ELSEPRINT"ERROR
:"ABS(147637-C):END
60 FOR A=8192 TO 10240:READ D:PO
KE A,D:C=C+D:NEXT A
70 PRINT"SECTION 3 ";:IF C=40075
4 THEN PRINT"OK":ELSEPRINT"ERROR
:"ABS(400754-C)
80 END
90 REM **** SECTION 1 ****
100 REM
110 DATA 106,107,108,109,110,111
,112,113,114,115,116,117,118,119
,120,121
120 DATA 122,123,124,125,126,127
,192,219,220,221,222,223,224,225
,226,227
130 DATA 228,229,230,231,232,233
,234,235,236,237,238,239,240,241
,242,243
140 DATA 244,245,246,247,248,249
,250,251,252,253,254,255
150 REM
160 REM **** SECTION 2 ****
170 REM
180 DATA 169,0,141,25,255,169,0,
133,208,169,12,133,209,169,64,16
0
190 DATA 0,162,4,145,208,200,208
,251,230,209,202,208,246,96,32,4
4
200 DATA 147,201,39,144,5,162,23
,76,131,134,169,32,160,39,153,0
210 DATA 12,136,208,250,24,160,1
,162,0,32,240,255,165,34,164,35
220 DATA 32,136,144,162,96,142,0
,12,232,142,39,12,96,169,0,133
230 DATA 208,169,12,133,209,169,
64,133,210,169,16,133,211,162,4,
160
240 DATA 0,177,208,145,210,200,2
08,249,230,209,230,211,202,208,2
42,96
250 DATA 169,64,133,208,169,16,1
33,209,169,0,133,210,169
260 DATA 12,133,211,208,219,32,1
29,157,134,208,224,39,144,5,162,
14
270 DATA 76,131,134,32,129,157,1
34,209,224,24,176,242,32,129,157
,134
280 DATA 210,224,15,176,233,24,1
64,208,166,209,32,240,255,165,21
0,10
290 DATA 10,170,164,202,189,0,16
,145,200,200,189,1,16,145,200,13
6
300 DATA 24,165,200,105,40,133,2
00,144,2,230,201,189,2,16,145,20
0
310 DATA 200,189,3,16,145,200,96
,32,129,157,134,208,224,37,144,5
320 DATA 162,14,76,131,134,32,12
9,157,134,209,224,23,176,242,32,
129
330 DATA 157,134,210,24,138,101,
208,201,39,176,229,32,129,157,13
4,211
340 DATA 24,138,101,209,201,24,1
76,216,164,208,166,209,32,240,25
5,165
350 DATA 202,101,200,133,200,144
,2,230,201,169,98,160,0,145,200,
169
360 DATA 100,230,210,164,210,145
,200,136,169,99,145,200,136,208,
251,166
370 DATA 211,232,208,12,160,0,16
9,101,145,200,164,210,169,102,14
5,200
380 DATA 24,165,200,105,40,133,2
00,144,2,230,201,202,208,230,160
,0
390 DATA 169,103,145,200,164,210
,169,105,145,200,136,169,104,145
,200,136
400 DATA 208,251,230,209,165,208
,141,231,7,24,101,210,56,233,2,1
41
410 DATA 232,7,230,209,165,209,1
41,230,7,101,211,56,233,2,141,22
9
420 DATA 7,169,147,76,210,255,32
,129,157,138,224,40,144,5,162,14
430 DATA 76,131,134,170,133,216,
32,129,157,138,224,24,176,240,13
3,217
440 DATA 96,24,164,216,166,217,3
2,240,255,24,165,202,101,200,133
,200
450 DATA 144,2,230,201,160,0,177
,200,141,194,247,201,64,240,6,16
9
460 DATA 91,145,200,208,4,169,93
,145,200,160,40,177,200,141,195,
247
470 DATA 201,64,240,6,169,92,145
,200,208,4,169,94,145,200,96,24
480 DATA 164,216,166,217,32,240,
255,120,141,63,255,24,165,202,10
1,200
490 DATA 133,200,144,2,230,201,1
60,0,173,194,247,145,200,160,40,
173
500 DATA 195,247,145,200,141,62,
255,88,96,32,129,157,142,195,22,
96
510 DATA 32,191,22,32,159,255,32
,228,255,240,248,133,208,201,84,
208
520 DATA 1,96,165,208,201,68,208
,12,165,216,240,231,32,48,22,198
530 DATA 216,76,97,22,165,208,20
1,54,208,14,165,216,201,39,240,2
11
540 DATA 32,48,22,230,216,76,97,
22,165,208,201,53,208,12,165,217
550 DATA 240,193,32,48,22,198,21
7,76,97,22,165,208,201,82,208,17
9
560 DATA 165,217,201,23,240,173,
32,48,22,230,217,76,97,22,32,242
570 DATA 21,160,40,162,0,232,208
,253,136,208,250,96,32,44,147,13
3
580 DATA 208,32,129,157,134,209,
224,4,144,5,162,14,76,131,134,32
590 DATA 129,157,138,224,28,176,
243,166,209,157,51,24,169,55,133
,211
600 DATA 169,24,133,212,165,209,
240,16,24,165,211,105,71,133,211
,144
610 DATA 2,230,212,198,209,76,24
5,22,164,208,169,0,145,211,136,1
77
620 DATA 34,145,211,136,208,249,
177,34,145,211,169,0,133,20,133,
21
630 DATA 169,25,133,22,169,0,133
,19,162,5,149,22,202,208,251,96
640 DATA 32,129,157,134,212,224,
4,144,5,162,14,76,131,134,166,21
2
650 DATA 189,51,24,133,208,169,1
,133,209,169,8,133,210,169,9,133
660 DATA 211,24,32,89,21,169,55,
133,208,169,24,133,209,165,212,2
40
670 DATA 16,24,165,208,105,71,13
3,208,144,2,230,209,198,212,76,9
4
680 DATA 23,160,0,177,208,240,16
,72,201,47,208,4,104,169,13,72
690 DATA 104,32,210,255,200,208,
236,162,5,189,45,24,32,210,255,2
02
700 DATA 208,247,166,101,188,51,
24,162,1,32,240,255,24,165,200,1
01
710 DATA 202,105,40,133,200,144,
2,230,201,169,2,133,217,169,160,
160
720 DATA 0,145,200,160,9,145,200
,136,177,200,9,128,145,200,136,2
08
730 DATA 247,32,40,24,32,228,255
,240,248,133,208,201,84,208,3,19
8
740 DATA 217,96,165,208,201,53,2
08,14,165,217,201,2,240,227,32,1
7
750 DATA 24,198,217,76,248,23,16
5,208,201,82,208,213,165,217,201
,10
760 DATA 240,207,32,17,24,230,21
7,166,101,188,51,24,166,217,24,3
2
770 DATA 240,255,24,165,200,101,
202,133,200,144,2,230,201,76,174
,23
780 DATA 160,0,169,101,145,200,1
60,9,169,102,145,200,136,177,200
,41
790 DATA 127,145,200,136,208,247
,96,32,194,22,76,159,255,84,73,8
8,197,13
800 REM
810 REM **** SECTION 3 ****
820 REM
830 DATA 0,60,102,110,110,96,60,
0,0,0,60,6,62,102,62,0
840 DATA 0,96,96,124,102,102,124
,0,0,0,60,102,96,102,60,0
850 DATA 0,6,6,62,102,102,62,0,0
,0,60,102,126,96,62,0
860 DATA 0,28,54,120,48,48,48,0,
0,0,62,102,102,62,6,124
870 DATA 0,96,96,124,102,102,102
,0,0,24,0,24,24,24,24,0
880 DATA 0,6,0,6,6,6,102,60,0,96
,96,108,120,124,102,0
890 DATA 0,56,24,24,24,24,60,0,0
,0,107,127,127,99,99,0
900 DATA 0,0,124,102,102,102,102
,0,0,0,60,102,102,102,60,0
910 DATA 0,0,124,102,102,124,96,
96,0,0,62,102,102,62,6,6
920 DATA 0,0,124,102,96,96,96,0,
0,0,60,96,60,6,124,0
930 DATA 0,48,48,124,48,54,28,0,
0,0,102,102,102,102,60,0
940 DATA 0,0,102,102,102,60,24,0
,0,0,99,107,127,54,34,0
950 DATA 0,0,102,60,24,60,102,0,
0,0,102,102,102,62,6,124
960 DATA 0,0,126,12,24,48,126,0,
0,60,48,48,48,48,60,0
970 DATA 0,28,54,120,48,54,124,0
,0,60,12,12,12,12,60,0
980 DATA 0,24,60,126,24,24,24,0,
```

```
0,16,48,126,126,48,16,0
990 DATA 0,0,0,0,0,0,0,0,0,24,24
,24,24,0,24,0
1000 DATA 0,108,108,108,0,0,0,0,
0,54,127,54,127,54,0,0
1010 DATA 24,62,104,60,22,124,24
,0,0,98,100,8,16,38,70,0
1020 DATA 0,60,102,60,103,102,63
,0,0,12,24,0,0,0,0,0
1030 DATA 0,24,48,48,48,48,24,0,
0,24,12,12,12,12,24,0
1040 DATA 0,102,60,126,60,102,0,
0,0,24,24,126,24,24,0,0
1050 DATA 0,0,0,0,24,24,48,0,0,0
,0,126,0,0,0,0
1060 DATA 0,0,0,0,0,24,24,0,0,6,
12,24,48,96,0,0
1070 DATA 0,60,102,110,118,102,6
0,0,0,24,56,24,24,24,126,0
1080 DATA 0,60,102,12,24,48,126,
0,0,60,102,28,6,102,60,0
1090 DATA 0,12,28,44,126,12,12,0
,0,126,96,124,6,102,60,0
1100 DATA 0,60,96,124,102,102,60
,0,0,126,6,12,12,24,24,0
1110 DATA 0,60,102,60,102,102,60
,0,0,60,102,102,62,6,60,0
1120 DATA 0,0,24,0,0,24,0,0,0,0,
24,0,24,24,48,0
1130 DATA 0,24,48,96,48,24,0,0,0
,0,126,0,126,0,0,0
1140 DATA 0,24,12,6,12,24,0,0,0,
60,102,12,24,0,24,0
1150 DATA 170,85,170,85,170,85,1
70,85,0,60,102,102,126,102,102,0
1160 DATA 0,124,102,124,102,102,
124,0,0,60,102,96,96,102,60,0
1170 DATA 0,124,102,102,102,102,
124,0,0,126,96,120,96,96,126,0
1180 DATA 0,126,96,120,96,96,96,
0,0,60,102,96,110,102,60,0
1190 DATA 0,102,102,126,102,102,
102,0,0,60,24,24,24,24,60,0
1200 DATA 0,30,12,12,12,108,56,0
,0,102,108,120,120,108,102,0
1210 DATA 0,96,96,96,96,96,126,0
,0,99,119,127,107,99,99,0
1220 DATA 0,102,118,126,110,102,
102,0,0,60,102,102,102,102,60,0
1230 DATA 0,124,102,102,124,96,9
6,0,0,60,102,102,102,60,14,0
1240 DATA 0,124,102,102,124,102,
102,0,0,60,96,60,6,102,60,0
1250 DATA 0,126,24,24,24,24,24,0
,0,102,102,102,102,102,60,0
1260 DATA 0,102,102,102,102,60,2
4,0,0,99,99,107,127,119,99,0
1270 DATA 0,102,102,60,60,102,10
2,0,0,102,102,60,24,24,24,0
1280 DATA 0,126,12,24,48,96,126,
0,0,128,192,224,240,248,252,254
1290 DATA 240,216,152,12,12,6,6,
0,42,149,202,229,242,249,252,254
1300 DATA 242,217,154,77,172,86,
166,80,108,16,16,16,16,16,16,108
1310 DATA 248,224,192,128,128,0,
0,0,31,7,3,1,1,0,0,0
1320 DATA 255,128,128,128,128,12
8,128,128,255,0,0,0,0,0,0,0
1330 DATA 254,3,3,3,3,3,3,3,128,
128,128,128,128,128,128,128
1340 DATA 3,3,3,3,3,3,3,3,128,12
8,128,128,128,128,255,127
1350 DATA 0,0,0,0,0,0,255,255,3,
3,3,3,3,3,255,255
1360 DATA 170,127,194,127,192,64
,192,64,170,85,170,255,2,3,2,3
1370 DATA 192,64,192,64,192,64,2
55,85,2,3,2,3,2,3,254,85
1380 DATA 170,87,170,127,160,127
,176,85,170,245,42,255,2,255,6,8
5
1390 DATA 181,85,181,85,181,85,1
76,95,86,85,86,85,85,85,6,253
1400 DATA 170,127,160,110,160,11
1,160,111,170,213,106,85,74,125,
6,5
1410 DATA 160,111,160,111,160,96
,191,85,6,197,6,245,6,5,254,85
1420 DATA 170,95,176,85,176,86,1
76,117,170,253,10,173,10,237,10,
15
1430 DATA 208,159,128,128,192,12
7,170,85,11,249,1,41,3,255,170,8
5
1440 DATA 170,85,170,85,170,127,
224,64,170,85,170,85,170,255,3,1
1450 DATA 193,95,193,64,224,127,
170,85,193,253,193,1,3,255,170,8
5
1460 DATA 170,85,191,96,160,96,1
60,97,170,85,254,5,6,5,6,133
1470 DATA 255,195,153,145,145,15
9,195,255,255,255,195,249,193,15
3,193,255
1480 DATA 255,159,159,131,153,15
3,131,255,255,255,195,153,159,15
3,195,255
1490 DATA 255,249,249,193,153,15
3,193,255,255,255,195,153,129,15
9,193,255
1500 DATA 255,227,201,135,207,20
7,207,255,255,255,193,153,153,19
3,249,131
1510 DATA 255,159,159,131,153,15
3,153,255,255,231,255,231,231,23
1,231,255
1520 DATA 255,249,255,249,249,24
9,153,195,255,159,159,147,135,13
1,153,255
1530 DATA 255,199,231,231,231,23
1,195,255,255,255,148,128,128,15
6,156,255
1540 DATA 255,255,131,153,153,15
3,153,255,255,255,195,153,153,15
3,195,255
1550 DATA 255,255,131,153,153,13
1,159,159,255,255,193,153,153,19
3,249,249
1560 DATA 255,255,131,153,159,15
9,159,255,255,255,195,159,195,24
9,131,255
1570 DATA 255,207,207,131,207,20
1,227,255,255,255,153,153,153,15
3,195,255
1580 DATA 255,255,153,153,153,19
5,231,255,255,255,156,148,128,20
1,221,255
1590 DATA 255,255,153,195,231,19
5,153,255,255,255,153,153,153,19
3,249,131
1600 DATA 255,255,129,243,231,20
7,129,255,255,195,207,207,207,20
7,195,255
1610 DATA 255,227,201,135,207,20
1,131,255,255,195,243,243,243,24
3,195,255
1620 DATA 255,231,195,129,231,23
1,231,255,255,239,207,129,129,20
7,239,255
1630 DATA 255,255,255,255,255,25
5,255,255,255,231,231,231,231,25
5,231,255
1640 DATA 255,147,147,147,255,25
5,255,255,255,201,128,201,128,20
1,255,255
1650 DATA 231,193,151,195,233,13
1,231,255,255,157,155,247,239,21
7,185,255
1660 DATA 255,195,153,195,152,15
3,192,255,255,243,231,255,255,25
5,255,255
1670 DATA 255,231,207,207,207,20
7,231,255,255,231,243,243,243,24
3,231,255
1680 DATA 255,153,195,129,195,15
3,255,255,255,231,231,129,231,23
1,255,255
1690 DATA 255,255,255,255,231,23
1,207,255,255,255,255,129,255,25
5,255,255
1700 DATA 255,255,255,255,255,23
1,231,255,255,249,243,231,207,15
9,255,255
1710 DATA 255,195,153,145,137,15
3,195,255,255,231,199,231,231,23
1,129,255
1720 DATA 255,195,153,243,231,20
7,129,255,255,195,153,227,249,15
3,195,255
1730 DATA 255,243,227,211,129,24
3,243,255,255,129,159,131,249,15
3,195,255
1740 DATA 255,195,159,131,153,15
3,195,255,255,129,249,243,243,23
1,231,255
1750 DATA 255,195,153,195,153,15
3,195,255,255,195,153,153,193,24
9,195,255
1760 DATA 255,255,231,255,255,23
1,255,255,255,255,231,255,231,23
1,207,255
1770 DATA 255,231,207,159,207,23
1,255,255,255,255,129,255,129,25
5,255,255
1780 DATA 255,231,243,249,243,23
1,255,255,255,195,153,243,231,25
5,231,255
1790 DATA 160,97,151,97,160,127,
170,85,255,195,153,153,129,153,1
53,255
1800 DATA 255,131,153,131,153,15
3,131,255,255,195,153,159,159,15
3,195,255
1810 DATA 255,131,153,153,153,15
3,131,255,255,129,159,135,159,15
9,129,255
1820 DATA 255,129,159,135,159,15
9,159,255,255,195,153,159,145,15
3,195,255
1830 DATA 255,153,153,129,153,15
3,153,255,255,195,231,231,231,23
1,195,255
1840 DATA 255,225,243,243,243,14
7,199,255,255,153,147,135,135,14
7,153,255
1850 DATA 255,159,159,159,159,15
9,129,255,255,156,136,128,148,15
6,156,255
1860 DATA 255,153,137,129,145,15
3,153,255,255,195,153,153,153,15
3,195,255
1870 DATA 255,131,153,153,131,15
9,159,255,255,195,153,153,153,19
5,241,255
1880 DATA 255,131,153,153,131,15
3,153,255,255,195,159,195,249,15
3,195,255
1890 DATA 255,129,231,231,231,23
1,231,255,255,153,153,153,153,15
3,195,255
1900 DATA 255,153,153,153,153,19
5,231,255,255,156,156,148,128,13
6,156,255
1910 DATA 255,153,153,195,195,15
3,153,255,255,153,153,195,231,23
1,231,255
1920 DATA 255,129,243,231,207,15
9,129,255,6,133,134,133,6,253,17
0,85
1930 DATA 170,85,170,85,255,64,1
92,91,170,85,170,85,254,3,2,219
1940 DATA 219,64,195,58,255,85,1
70,85,218,3,194,35,254,85,170,85
1950 DATA 191,96,223,80,223,64,2
05,77,254,7,250,11,250,3,178,179
```

```
1960 DATA 192,77,205,64,205,77,2
24,127,2,179,178,3,178,179,6,253
1970 DATA 196,77,139,156,240,97,
161,65,186,101,162,115,30,13,10,
5
1980 DATA 193,66,164,96,176,124,
251,245,6,5,10,13,26,125,190,95
1990 DATA 255,143,191,143,175,17
5,167,135,255,255,195,129,153,15
3,153,153
2000 DATA 131,147,143,175,190,15
9,143,255,129,153,153,129,7,255,
255,255
2010 DATA 255,143,175,159,175,17
5,167,135,255,255,195,129,189,13
3,133,157
2020 DATA 131,147,143,143,158,14
3,143,255,145,129,145,129,7,255,
255,255
2030 DATA 255,146,255,0,255,170,
255,213,255,79,255,0,255,171,255
,85
2040 DATA 255,170,255,213,255,0,
0,0,255,171,255,85,255,0,0,0
2050 DATA 0,3,4,5,7,3,0,1,0,192,
224,224,224,192,0,128
2060 DATA 1,7,0,31,63,0,63,31,12
8,224,0,248,252,0,252,248
2070 DATA 0,0,1,1,3,3,6,4,0,0,12
8,128,192,192,224,224
2080 DATA 12,9,31,16,48,32,112,0
,112,112,248,56,60,28,62,0
2090 DATA 0,0,3,26,22,18,11,9,0,
0,0,192,160,88,108,42
2100 DATA 101,84,76,32,16,8,6,4,
170,2,2,4,4,8,8,8,0
```

SIMPLE

PROGRAM: SIMPLE DEMO

```
10 IF C=0 THEN C=1:LOAD"SIMPLE",
8,1
20 DESK=5204:MENU=5234:SRE=5281:
FETCH=5316:ICN=5334:WIND=5416:PS
TN=5591
30 SHOW=5618:HIDE=5680:SPEED=572
2:MOVE=5729:MAKE=5837:PULL=5937:
ENABLE=6485
40 DIM I$(11):SYS ENABLE:SYS DES
K:SYS MENU "[s S]CREEN [s P]OINT
ER [s D]ATA [s G]OODIES"
50 SYS MAKE "[s D]ESK/[s M]ENU/[
s I]CON/[s W]INDOW",0,1
60 SYS MAKE "[s P]OSITION/[s S]P
EED/[s S]HOW/[s H]IDE/[s M]OVE",
1,8
70 SYS MAKE "[s S]TORE/[s F]ETCH
/[s M]AKE/[s P]ULL/[s A]DDRESS/[
s I][s R][s Q]",2,16
80 SYS MAKE "[s I]NPUT/[s S]PEED
/[s I]NFO/[s Q]UIT",3,21
90 SYS WIND,1,2,36,1:PRINTSPC(9)
"[s T]HE '[s S][s I][s M][s P][s
 L][s E]' [s D]EMO";:FOR I=0 TO
11:READ I$(I):NEXT
100 FOR I=0 TO 8:SYS ICN,(3+I*4)
,22,I:NEXT I:DEF FNP(D)=PEEK(307
2+Y*40+X)
110 S=4:SYS SPEED,S*10:SYS PSTN,
20,12
120 SYS MOVE:X=PEEK(216):Y=PEEK(
217)
130 IF Y=0 THEN 180
140 SYS HIDE:I=INT((X-3)/4)+1:SY
S SRE
150 IF FNP(0)=64 THEN I=10:ELSEI
F Y<22 THEN I=0
160 SYS WIND,9,11,20,1:PRINT"[s
T]HE "I$(I);
170 TI$="000000":DO:LOOPUNTIL TI
$="000001":SYS FETCH:GOTO 120
180 SYS HIDE:IF FNP(0)=32 OR X=0
 OR X=39 THEN SYS SRE:I=11:GOTO
160
190 IF X<7 THEN M=0:ELSE IF X<16
 THEN M=1:ELSE IF X<20 THEN M=2:
ELSE M=3
200 SYS SRE:SYS PULL,M:SYS FETCH
:O=PEEK(217)
210 ON M+1 GOTO 270,520,820,1170
220 GOTO 120
230 END
240 DATA "[s T]ITLE [s W]INDOW",
"[s F]ILE [s I]CON","[s T]RASH [
s C]AN","[s D]OCUMENT [s I]CON",
"[s P]RINTER [s I]CON"
250 DATA "[s D]ISK-[s D]RIVE [s
I]CON","[s D]ISK [s I]CON","[s C
]ASSETTE [s I]CON","[s C]ALCULAT
OR [s I]CON","[s C]LOCK [s I]CON
"
260 DATA "[s D]ESK-TOP","[s M]EN
U [s H]EADER"
270 IF O>4 THEN 120
280 SYS WIND,1,9,36,10:ON O GOSU
B 320,370,430,470
290 CHAR 1,18,19,"[RVSON] [s O][
s K] [RVSOFF]":SYS PSTN,23,18
300 SYS MOVE:X=PEEK(216):Y=PEEK(
217):SYS HIDE:IF Y<18 OR Y>19 TH
EN 300
310 IF X<18 OR X>21 THEN 300:ELS
E SYS FETCH:GOTO 120
320 PRINTSPC(15)"[RVSON] [s D][s
 E][s S][s K] [RVSOFF]":PRINT"[D
OWN][s S]YNTAX:  [s S][s Y][s S]
 5204"
330 PRINT"[DOWN][s F]UNCTION: [s
 C]LEARS [s S]CREEN TO THE GREY"
340 PRINTSPC(10)"DESKTOP PATTERN
 COMMONLY"
350 PRINTSPC(10)"USED IN MOUSE-D
RIVEN":PRINTSPC(10)"PACKAGES."
360 RETURN
370 PRINTSPC(15)"[RVSON] [s M][s
 E][s N][s U] [RVSOFF]":PRINT"[D
OWN][s S]YNTAX: [s S][s Y][s S]
5234 "CHR$(34)"[s M]ENU [s H]EAD
ING"CHR$(34)
380 PRINT"[DOWN][s F]UNCTION: [s
 P]RINTS [s H]EADING FOR THE"
390 PRINTSPC(10)"[s P]ULL-[s D]O
WN MENUS. [s Y]OU MAY"
400 PRINTSPC(10)"HAVE UP TO FOUR
 MENUS BUT"
410 PRINTSPC(10)"THE TEXT CAN SA
Y ANYTHING."
420 RETURN
430 PRINTSPC(15)"[RVSON] [s I][s
 C][s O][s N] [RVSOFF]":PRINT"[D
OWN][s S]YNTAX: [s S][s Y][s S]
5334,X,Y,ICON"
440 PRINT"[DOWN][s F]UNCTION: [s
 P]RINTS AN ICON 0-14 AT"
450 PRINTSPC(10)"THE COORDINATES
 (X,Y)."
460 RETURN
470 PRINTSPC(14)"[RVSON] [s W][s
 I][s N][s D][s O][s W] [RVSOFF]
":PRINT"[DOWN][s S]YNTAX: [s S][
s Y][s S] 5416,X,Y,WIDTH,HEIGHT"
480 PRINT"[DOWN][s F]UNCTION: [s
 D]RAWS AND CREATES A"
490 PRINTSPC(10)"WINDOW AT (X,Y)
. WITH THE"
500 PRINTSPC(10)"GIVEN WIDTH AND
 HEIGHT."
510 RETURN
520 O=PEEK(217):IF O>5 THEN SYS
FETCH:GOTO 120
530 SYS WIND,1,9,36,10:ON O GOSU
B 550,590,650,700,760
540 GOTO 290
550 PRINTSPC(13)"[RVSON] [s P][s
 O][s S][s I][s T][s I][s O][s N
] [RVSOFF]":PRINT"[DOWN][s S]YNT
AX: [s S][s Y][s S] 5591,X,Y"
560 PRINT"[DOWN][s F]UNCTION: [s
 M]OVES [s P]OINTER TO THE"
570 PRINTSPC(10)"COORDINATES (X,
Y)."
580 RETURN
590 PRINTSPC(15)"[RVSON] [s S][s
 P][s E][s E][s D] [RVSOFF]":PRI
NT"[DOWN][s S]YNTAX: [s S][s Y][
s S] 5722,SPEED"
600 PRINT"[DOWN][s F]UNCTION: [s
 S]ETS SPEED OF MOVEMENT"
610 PRINTSPC(10)"FOR THE POINTER
. [s T]HE"
620 PRINTSPC(10)"HIGHER THE VALU
E, THE"
630 PRINTSPC(10)"LOWER THE SPEED
."
640 RETURN
650 PRINTSPC(15)"[RVSON] [s S][s
 H][s O][s W] [RVSOFF]":PRINT"[D
OWN][s S]YNTAX: [s S][s Y][s S]
5618"
660 PRINT"[DOWN][s F]UNCTION: [s
 S]AVES CHARACTERS AT THE"
670 PRINTSPC(10)"POINTER COORDIN
ATES AND"
680 PRINTSPC(10)"DRAWS THE POINT
ER."
690 RETURN
700 PRINTSPC(15)"[RVSON] [s H][s
 I][s D][s E] [RVSOFF]":PRINT"[D
OWN][s S]YNTAX: [s S][s Y][s S]
5680"
710 PRINT"[DOWN][s F]UNCTION: [s
 R]EPLACES THE CHARACTERS"
720 PRINTSPC(10)"HIDDEN BY THE P
OINTER AND"
730 PRINTSPC(10)"REMOVES IT FROM
 THE"
740 PRINTSPC(10)"SCREEN."
750 RETURN
760 PRINTSPC(15)"[RVSON] [s M][s
 O][s V][s E] [RVSOFF]":PRINT"[D
OWN][s S]YNTAX: [s S][s Y][s S]
5729"
770 PRINT"[DOWN][s F]UNCTION: [s
 A]LLOWS YOU TO MOVE THE"
780 PRINTSPC(10)"POINTER AROUND
THE SCREEN"
790 PRINTSPC(10)"UNTIL THE <[s R
][s E][s T][s U][s R][s N]> KEY
IS"
800 PRINTSPC(10)"PRESSED."
810 RETURN
820 IF O>6 THEN SYS FETCH:GOTO 1
20
830 SYS WIND,1,9,36,10:ON O GOSU
B 850,890,930,990,1030,1110
840 GOTO 290
850 PRINTSPC(15)"[RVSON] [s S][s
 T][s O][s R][s E] [RVSOFF]":PRI
NT"[DOWN][s S]YNTAX: [s S][s Y][
s S] 5281"
860 PRINT"[DOWN][s F]UNCTION: [s
 S]TORES THE CURRENT SCREEN"
870 PRINTSPC(10)"IN MEMORY."
880 RETURN
```

```
890 PRINTSPC(15)"[RVSON] [s F][s
 E][s T][s C][s H] [RVSOFF]":PRI
NT"[DOWN][s S]YNTAX: [s S][s Y][
s S] 5316"
900 PRINT"[DOWN][s F]UNCTION: [s
 R]ECALLS SCREEN FROM [s R][s A]
[s M]"
910 PRINTSPC(10)"AND COPIES TO S
CREEN."
920 RETURN
930 PRINTSPC(15)"[RVSON] [s M][s
 A][s K][s E] [RVSOFF]":PRINT"[D
OWN][s S]YNTAX: [s S][s Y][s S]
5837 "CHR$(34)"[s O]P1/[s O]P2/"
CHR$(34)",MENU,TAB"
940 PRINT"[DOWN][s F]UNCTION: [s
 C]REATES A PULL-DOWN MENU."
950 PRINTSPC(10)"[s T]AB IS THE
X-COORDINATE AT"
960 PRINTSPC(10)"WHICH THE MENU
(0-3) IS TO"
970 PRINTSPC(10)"BE PULLED DOWN.
"
980 RETURN
990 PRINTSPC(15)"[RVSON] [s P][s
 U][s L][s L] [RVSOFF]":PRINT"[D
OWN][s S]YNTAX: [s S][s Y][s S]
5937,MENU"
1000 PRINT"[DOWN][s F]UNCTION: [
s P]ULLS MENU DOWN AND ALLOWS"
1010 PRINTSPC(10)"AN OPTION TO B
E SELECTED."
1020 RETURN
1030 PRINTSPC(13)"[RVSON] [s A]D
DRESSES [RVSOFF]"
1040 PRINT"[DOWN]$[s D]8 - [s P]
OINTER [s X]-[s C]OORDINATE"
1050 PRINT"$[s D]9 - [s P]OINTER
 [s X]-[s C]OORDINATE"
1060 PRINT"$[s D]9 - [s O]PTION
[s S]ELECTED ([s A]FTER [s P][s
U][s L][s L])"
1070 PRINT"$[s C]8-$[s C]9 - [s
P]OINTER [s S]CREEN [s A]DDRESS"
1080 PRINT"$2000 - [s F]ONT [s S
]TART [s A]DDRESS"
1090 PRINT"$2801 - [s B][s A][s
S][s I][s C] [s S]TART [s A]DDRE
SS"
1100 RETURN
1110 PRINTSPC(15)"[RVSON] [s I][
s R][s Q] [RVSOFF]":PRINT"[DOWN]
[s S]YNTAX: [s S][s Y][s S] 6485
 - [s O][s N]"
1120 PRINTSPC(8)"[s S][s Y][s S]
 6498 - [s O][s F][s F]"
1130 PRINT"[DOWN][s F]UNCTION: [
s T]URNS ON OR OFF THE"
1140 PRINTSPC(10)"INTERRUPT WHIC
H RESETS THE"
1150 PRINTSPC(10)"FONT POINTERS.
"
1160 RETURN
1170 IF O>4 THEN SYS FETCH:GOTO
120
1180 ON O GOTO 1260,1340,1190,13
30
1190 SYS WIND,1,9,36,10:PRINTSPC
(14)"[RVSON] [s S][s I][s M][s P
][s L][s E] [RVSOFF]"
1200 PRINT"[DOWN][s S][s I][s M]
[s P][s L][s E] WAS WRITTEN IN J
UST THREE"
1210 PRINT"DAYS OF [s D]ECEMBER
1987 BY [s M]ARK"
1220 PRINT"[s E]VERINGHAM. [s T]
HANKS GO TO [s P]AUL [s D]IAS"
1230 PRINT"WHO PROVIDED ME WITH
THE ICONS."
1240 GOTO 290
1250 SYS WIND,1,9,36,1:OPEN1,0
1260 SYS WIND,1,6,5,2:SYS ICN,2,
7,11:SYS ICN,5,7,12:SYS PSTN,2,7
1270 SYS MOVE:SYS HIDE:X=PEEK(21
6):Y=PEEK(217)
1280 IF X<2 OR X>6 OR X=4 OR Y<7
 OR Y>8 THEN 1270:ELSE IF X<5 TH
EN 1310
1290 POKE 5743,84:POKE 5750,68:P
OKE 5768,54:POKE 5788,53:POKE 58
06,82
1300 POKE 6093,84:POKE 6102,53:P
OKE 6122,82:SYS FETCH:GOTO 110
1310 POKE 5743,13:POKE 5750,157:
POKE 5768,29:POKE 5788,145:POKE
5806,17
1320 POKE 6093,13:POKE 6102,145:
POKE 6122,17:SYS FETCH:GOTO 110
1330 PRINTCHR$(27)"R[s B][s A][s
 S][s I][s C]":END
1340 SYS SRE:SYS WIND,1,6,36,3:P
RINT"[DOWN][s S][s P][s E][s E][
s D]: -----------------------
":SYS PSTN,11,8
1350 CHAR 1,35-S,7,"[RVSON] [DOW
N][LEFT] [DOWN][LEFT] [RVSOFF]"
1360 SYS MOVE:X=PEEK(216):Y=PEEK
(217):SYS HIDE:IF Y<7 OR Y>9 OR
X<10 OR X>34 THEN 1360
1370 S=25-(X-10):SYS SPEED,S*10:
SYS FETCH:GOTO 120
```

YOUR COMMODORE

Lineage: 58p per word. (including VAT)
Semi display: £10.95 plus VAT per single column centimetre minimum 2cm. Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699

Send your requirements to:
ALAN COLE
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1R 3AB.

DISCS

DISKART

READY-TO-USE
GEOS GRAPHICS DISKS

Attention GEOS users! Tired of going through the cost and expense of converting other graphics so the GEOS format? Here are disks or ORIGINAL graphics that take advantage of the great GEOS operating system! Created with GEOS for GEOS each of these disks provide a selection of graphics that will tickle the eye (and the wallet).

DISKART 1 (assorted)		
Graphics Goodies 1	Weather Stuff 1	Graphic Goodies 2
Holidays 1	GEOpaint Tips 1	GEOpaint Tips 2
DISKART 2 (assorted)		
Graphic Goodies 3	U.S. Map 1	Little Guys 1
Workdisk Labels	Holidays 2	GEOpaint Tips 1
Musical Stuff 1		
DISKART 3 (wheels and wings)		
Vehicles 1	Vehicles 2	DC-3 Airliner
PORSCHE 959	F4 Phantom	Warbirds 1
Tin Lizzies	Nieuport 17	
DISKART 4 (assorted)		
Little Guys	Ovals/ Blocks/ etc	Foodstuff 1
C64 and Periphs	Tools 1	Spring/ Summer stuff
DISKART 5 (assorted)		
Make-A-Face	Drafting Equip	Vehicles 3
Banners 1	Flowchart Symbols	Little Women 1
Little Women 2	Houses 1	
DISKFORMS (5 different, blank, forms, plus 3 ready-to-print forms)		
INSTRUCTIONS	Lined Paper	Delivery Receipt
Blank Form 1	Inventory Form	Bank Deposit Form
Blank Form 2	Blank Form 3	GEOpaint Tips 1
Blank Form 4	Blank Form 5	
MUSI-KIT (create professional-looking sheet music)		
Musi-Kit Info	Piano Title	Piano Stuff
Musi-Kit	Single Title	Single Staff
Large Instruments	Sample Sheet	

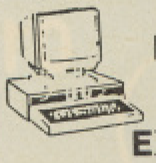
TO ORDER: Send **£9.95** for each disk. To: **Financial Systems Software Ltd.**, 2nd Floor, Anbrian House, St. Mary's Street, Worcester WR1 1HA. Tel: **(0386) 750217** (9.30 to 8.30 pm) GEOS is a registered trade mark of Berkeley Softworks.

SPECIAL OFFERS

For details of Autumn offers watch this space or call
0395 267734
EXMOUTH MICROS
13 South Street,
Exmouth, Devon.
EX8 2SX.

DISCOUNT SOFTWARE: Latest titles for CBM64/128 at 75% off R.R.P.: S.A.E. To: A.O.W. 104 Hollywell Road, Lincoln. LN5 9BY.

MODEM for CBM 64/VIC 20
Switchable CCITT V21/Bell 103, ORIGINATE or ANSWER modes. No external power required. Supplied with cable to connect to user port. Non/approved.
Price **£39.00** (all inclusive)
INTELNET LTD, Unit C2, 8-12 Creekside, London SE8 3DX



MARKET YOUR EXPERTISE IN YOUR COMMODORE CALL ALAN COLE ON 01-437 0699 TODAY!

SOFTWARE

YOUR COMMODORE SPECIALS

TECH DRAW 64 - A comprehensive technical illustration aid for C64 owners for both disk and tape. Commands available include LINE DRAW, RAYS, BOX, CIRCLE, ARC, ELLIPSE, FREEHAND, FILL, HATCH & TEXT. Wipe copy, move and rotate, save and load or print your drawing all for £8.95 tape or disk. Order Codes TD64T (tape) TD64D (disk)

Full details in Your Commodore July 1987.

SPEEDY ASSEMBLER

Your Commodore's very own Assembler, a 100% memory resident program for loading from tape or disk, featured in the Your Commodore Machine Code Series and now the Y.C. standard assembler only £9.95 Tape or Disk. Order Codes YSPASSD (disk) YSPASSR (tape)

Order from Your Commodore Readers Services at 9, Hall Road, Maylands Wood Estate, Hemel, Hempstead Herts HP2 7BH. Please make cheques payable to A.S.P. Ltd., or telephone your Access/Visa order on 0442 41221

X RATED
ADULTS ONLY GAME
FANTASY
Commodore 64/128 **£6.50**
inc p&p
Available to persons over 18 only. Please state age when ordering.
R'n'H MICROTEC
32 Hazell Way, Stoke Poges, Bucks SL2 4DD.

C64 GAME WRITERS

Give your games a professional finish with our copyright *FREE* packages. Disks/Tape incl.
LT1;; 300 Sprites (50 Objects) **£4.95**
LT2; 50 SFX (1,2,,,&3 Channel) **£4.49**
LT3; 3 Soundtracks (Title, In-Game, Game Over, Hi-Score & M15) **£5.79**
Full instructions incl.
Cheques P/O to:-
LATERAL THOUGHT
PO Box 418, Rutherglen, Glasgow G73 2EY.

FREE



Deluxe 120 Disk Capacity Lockable Disk box when you order
25 Disks for only **£18.25**
OR 50 Disks for only **£27.50**
OR 100 Disks for only **£44.50**

FREE CATALOGUE with every order

NEBULAE, Dept YC, FREEPOST, Kilroot Park Industrial Estate, Carrickfergus, Co. Antrim BT38 7BR Tel: (09603) 51020

Small business Accounts?

Doing your accounts is simple using **"Page-Accounting"**. It's a word processor that will do all your usual letters and labels plus a lot extra! You can keep a Cash Book, Bank account, Sales/Purchase Day books-ledger. Even produce a trial balance and a cashflow. Yourbank manager and accountant will be delighted.
C64/C128 **£24.95**, Amiga **£34.95**, inc VAt & P&P. Details from:-
FIELDMASTER SOFTWARE CO., Dept YCM1, 107 Oakwood Park Road, London N14 6QD Tel: 01-886-6709

SERVICES

C64 C128
READIDATA
The Commodore Specialists For Repairs
C16 Call 01-861-2835 +4

C64 POWER SUPPLIES
£13.50 inc. P&P
Refurbished 6 month guarantee.
Cheques/Postal Orders.
To:- B. Thompson, 23 Curzon Street, Derby.
Tel: 0332 291 219
Trade Enquiries Welcome.

BUSINESS OPPORTUNITIES

IDEAS, INVENTIONS Wanted. Call I.S.C. 01 434 1272 or write: Dept (ASP) 99 Regent St, London W1.

LANGUAGES

PLUS4 FORTH DISK. £24.95

MetacomPiler, FIG/79. 32 Kbytes.
Auto Tape/disk and Start.
HIRES: DRAW & DUMP. DEFINABLE CHARS: Complete with 50+ commented screens.
TENSAL TECHNOLOGY
15, Penrice Close, W-S-M. Avon. BS22 9AH

MODEM USERS

IT IS ALWAYS FREE TO LOOK AT DATAVISION - KEY
01-833 3799

NEXT COPY DEADLINE FOR OCTOBER ISSUE IS 21st JULY DON'T MISS IT.

TERMS

CLASSIFIED ADVERTISING TERMS & CONDITIONS

Our terms for new advertisers (semi-display and lineage) are strictly pro-forma payments until satisfactory reference can be taken up (excluding recognised advertising agencies). Cheques and P.O.'s should be crossed and made payable to ARGUS SPECIALIST PUBLICATIONS LTD and sent together with the advertisements to:

The Classified Dept.,
No. 1 Golden Square,
London W1R 3AB.

There are no reimbursements for cancellations. Advertisements arriving too late for a particular issue will be inserted in the following issue unless accompanied by instructions to the contrary. It is the responsibility of the advertiser to ensure that the first insertion of every series is published correctly, and corrections must be notified in time for the second insertion, otherwise the publishers will not accept liability or offer any reduction in charges.

All advertising sales are subject to Government Regulations concerning VAT. Advertisrs are responsible for complying with the various legal requirements in force eg: The Trade Description Act, sex discrimination act & the business advertisements (disclosure) order 1977.

Full Terms & Conditions of Advertising available on ruquest.

B R E A K

Bug Finder

We'd like to remind our readers that we run a Bug Finder service.

If you have typed in one of our programs and despite much checking, you still can't get it to run, then send us the following:

Two copies of your program on tape or disk.
A description of your problem.
If possible a listing of your work (you may omit this).
A stamped, self-addressed envelope for return of the program to you.

Should any of the above be missing then we will not be able to deal with your query.

We will try to point out where you have made errors and place a corrected copy of the program back on to your tape or disk before we return it to you.

Do not send a program to us as soon as it stops working, please check it several times first.

We do get a large number of queries and so it may take a while for us to deal with yours personally.

Note: we can only deal with problems relating to programs published in *Your Commodore*.

Commodore Where Are You?

At the *Your Commodore* office we are repeatedly asked for the address and telephone number of Commodore U.K. Many people, after referring to their computer manuals, believe them to be based in Corby.

The Commodore plant at Corby was closed down some time ago. Reproduced here you will find the correct address for Commodore U.K.

We suggest that you write this correct address in the front of your computers manual for future reference.

Commodore Business Machine, (UK),
Commodore House,
The Switchback,
Gardner Road,
Maidenhead,
Berks SL6 7XA.

Oops

Correction to Easy Basic Toolkit (June '88).
Saver Program:
The last data item in line 90 should be 144 NOT 137.
Tape users should type POKE40715,1 after running.
All users should then type SYS40704 to start the save.
Code program: The following lines were miss-printed,
line 4340 DATA
32, 40, 186, 230, 253, 76,184,196,166,2,224, 4,208,11,32,59,1903
line 4430 DATA
166,20,32,205,189,169,52,160, 197,32,30,171,165,21,3,2,135,1776

At the *Your Commodore* office we receive hundreds of letters from readers every month. We do try and answer each individually but sometimes this is impossible due to pressure of work. If you have written to us and not received a personal reply, we apologise for this but we cannot promise to reply to every item of mail we receive. If you feel that your question or letter really needs an answer, then inclusion of an s.a.e. will guarantee a reply, although this may still take time to arrive.

Puzzle Corner

Complete either of our cryptic puzzles
and win a *Your Commodore* binder!

A Complete the following phrases. For example, 26 L of the A becomes 26 letters of the alphabet.

1. 12 M in a Y
2. 52 C in a P
3. 88 K on a P
4. 9 P in the SS
5. 92 T in the FL
6. 4 H of the A
7. 12 D of C
8. 11 P in a CT
9. 366 D in a LY
10. 12 M of a J
11. 50 S on the AF
12. 2 N in a B

B

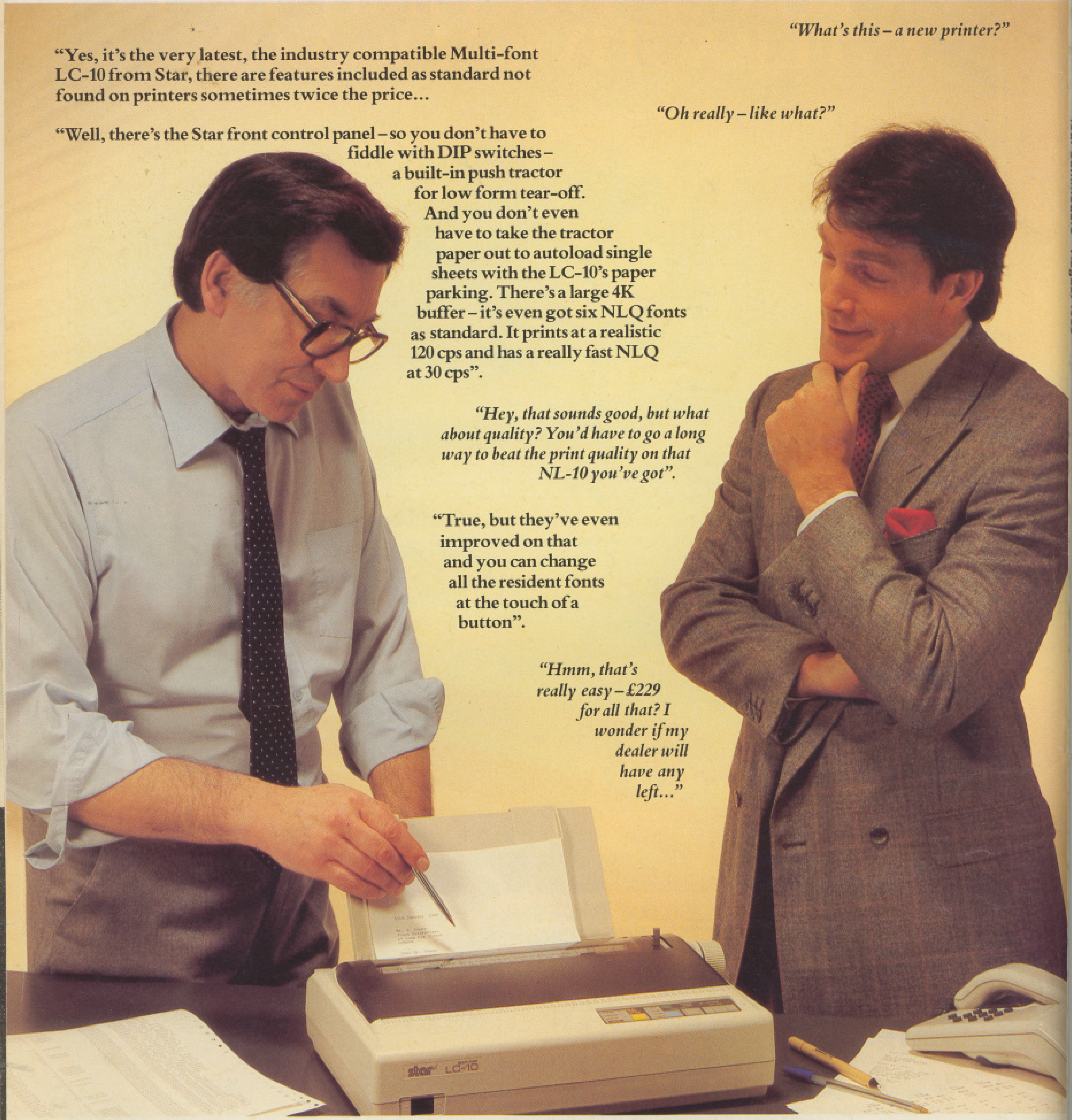
What is the next letter in the following series:

O,T,T,F,F,S,S ?

Mark clearly on envelope whether puzzle A or B and send to *your Commodore*, ASP Ltd, 1 Golden Square, London W1R 38B.


MICRONET. THE FIRST TRULY INTERACTIVE MAGAZINE.
COMMUNICATIONS
Celebrity Chatline
INFORMATION
JOB SEARCH
PC
PRISM Modem2000
THORN EMI Datatech

"£229 for all that?"



THE star PRINTERS FOR BUSINESS

Star Micronics U.K. Ltd.,
Craven House, 40 Uxbridge Road,
Ealing, London W5 2BS.
Telephone: 01-840 1800.

A division of
Star Micronics Co., Ltd., Japan.

Price excl VAT.

I'd like to find out if my dealer has any left. Please send me an LC-10 brochure ☐
Please send me details of the Star range of dot matrix printers ☐ *Or just call Belinda on 01-840 1829.*

Name ____________ *Company* ____________

Address ____________

____________ *Postcode* ____________

Telephone ____________

LCYC8

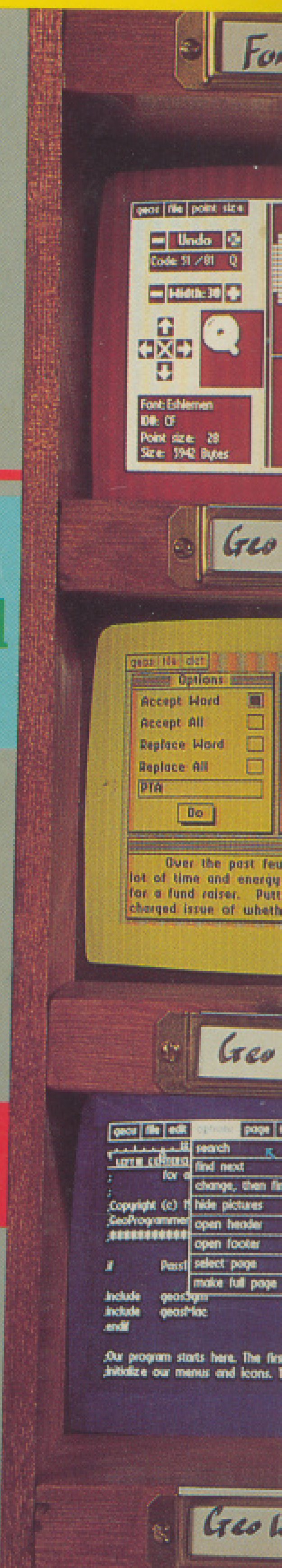
Your Commodore August 1988

Volume 4 Number 11

YOUR
COMM…

AUGUST 1988

GEOS–B…



Wordprocessors
Which one's for you?

Expert V3.2
As fast as a parallel DOS

Writing Structured Programs

UNBEATABLE PROGRAMS:

- Plus/4 SIMPLE
- C64/C128 Terminal
- Tape Organiser
- Educating SID
- A Short Interlude

Games Reviewed: Kar…
Wizard Warz ▲ I…